Impact Point Prediction and Track Fusion for Heterogeneous Sensors

Ting Yuan, Ph.D.
University of Connecticut, 2013

ABSTRACT

In this dissertation, two topics in the area of estimation, target tracking and data fusion are investigated: impact point prediction (IPP) and track fusion for heterogeneous sensors.

The first topic of interest is the problem of estimating the state of thrusting/ballistic projectiles for the end purpose of IPP. For an IPP algorithm using very short time observations from a single active or passive sensor, we face two major challenges. First, for a thrusting/ballistic projectile, different phases of its trajectory require different mathematical models to describe the corresponding physical behaviors — there is uncertainty as to which motion model is in effect. Secondly, in practical situations, for a trajectory with unknown drag coefficient and unknown thrust, it is difficult to separate between them given a very short observation time — there is an ambiguity in their estimation. Utilizing state-of-the-art techniques, we propose an IPP algorithm using state estimation

approach (multiple interacting multiple model estimators) with a single active sensor and an IPP algorithm using parameter estimation approach (maximum likelihood estimator) with a single passive sensor. These are further extended under more realistic and complicated environments, i.e., in the presence of wind and with no launch point information, respectively.

Secondly, track fusion for heterogeneous sensors will be presented. For distributed multisensor tracking systems, fusing local track estimates from multiple sensors into system tracks achieve better estimation performance than single sensor tracking. However, if one has a heterogeneous mix of sensors — active (with full position measurements) and passive (with line of sight measurements) — they use different system models in different state spaces. Compared with homogeneous track-to-track fusion that assumes the same system model for different sensors, the heterogeneous case poses two major challenges. The first one is that we have to fuse estimates from different state spaces (related by a nonlinear transformation). The second is the estimation errors are dependent due to the ”common process noise effect” and there is no known way to capture the “common” part exactly. The linear minimum mean squared error (LMMSE) and the maximum likelihood (ML) approaches are developed for this nonlinear case and compared with the corresponding centralized measurement tracker/fuser (CTF).

Impact Point Prediction and Track Fusion for Heterogeneous Sensors

Ting Yuan

B.S. & M.S., University of Electronic Science and Technology of China, 2006

M.S., University of Connecticut, 2012

A Dissertation
Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy
at the
University of Connecticut

2013

APPROVAL PAGE

Doctor of Philosophy Dissertation

Impact Point Prediction and Track Fusion for Heterogeneous Sensors

Presented by
Ting Yuan, B.S. ECE., M.S. ECE.

Major Advisor _____
Yaakov Bar-Shalom

Associate Advisor _____
Peter K. Willett

Associate Advisor _____
Krishna R. Pattipati

University of Connecticut
2013

ACKNOWLEDGEMENTS

My most humble and sincere thanks to:

- First and foremost, my major advisor Professor Yaakov Bar-Shalom, for his "full-time" excellent guidance and great sense of humor. From the tracking Maestro, I have learned how to KISS (keep it simple, stupid!) and the difference between ML (maximum likelihood) and SEX (state estimation of $\mathbf{x}$). Have to say, his superb instincts in research always motivate me to reach a higher level.
- Professor Peter Willett, my BABY (best advisor besides Yaakov), for his broad knowledge and inspirational expectoration-type jokes that always bring freshness. Professor Krishna Pattipati and Professor Shengli Zhou, for the knowledge I have learned in and out of the classroom will benefit the rest of my life.
- My labmates Xin Tian, Richard Osborne, III, Shuo Zhang, David Crouse, Djedjiga Belfadel, Kevin Romeo and Wenbo Dou, for their helpful discussions in research. My lab-neighbours Xiufeng Song, Jianzhong Huang, Zhaohui Wang, Ramona Georgescu and Sora Choi, for making my lunch time special.
- The CHWL organization distributed in Yibin, Chengdu and Shenzhen, for those joyful memories over the last few decades. The Hammer group in Connecticut, for those beers and wines, yin and yang throughout these many years.
- All the people I have loved, secretly or openly, for no particular reason.
- My parents Fawen Yuan and Zeqiong Guo, my uncle Guocai Jiang and aunt Zequn Guo, for their unconditional love and support. To my family I dedicate this dissertation.

TABLE OF CONTENTS

Introduction 1
0.1. Motivation 1
0.1.1. State Estimation for IPP with an Active Sensor 1
0.1.2. Parameter Estimation for IPP with a Passive Sensor 3
0.1.3. Heterogeneous Track-to-Track Fusion 5
0.2. Outline 7
0.3. Publications 9

Ch. 1 : MIMM-UM Estimation Approach for IPP 12
1.1. Introduction 12
1.2. IMM Estimator with Unbiased Mixing 15
1.3. System Model and its Discretized Form 20
1.4. Best Initial Drag Coefficient Estimate of MIMM Estimator 25
1.5. Impact Point Prediction 28
1.6. Real Data Results 29
1.6.1. MIMM Approach 29
1.6.2. *N*-point Initialization and Measurement Noise Intensities 30
1.6.3. Selection of Process Noise Intensities 33
1.6.4. IMM Mode Transition Probability Matrix 34
1.6.5. Real Data Results 35
1.7. Conclusions 42

Ch. 2 : IPP for Thrusting Projectiles in the Presence of Wind 45
2.1. Introduction 46
2.2. Dynamic Model in the Presence of Wind 49
2.3. Multiple-IMM Estimator for IPP 56
2.4. MIMM Design Parameter Selection 59
2.4.1. Selection of the Best IMM Estimator 60
2.4.2. *N*-point Adaptive Initialization 61
2.4.3. Selection of Process Noise Intensities 64
2.4.4. IMM Mode Transition Probability Matrix 66
2.5. Simulation Results 66
2.5.1. IPP RMS Error 69
2.5.2. IPP Uncertainty Ellipse 74
2.5.3. Consistency Evaluation 76
2.6. Summary and Conclusions 77

Ch. 3 : IPP in 3-D using 2-D Measurements from a Single Passive Sensor 83
3.1. Introduction 84

3.2. The Problem 87
3.3. Possible Ambiguity and the ML Estimator 93
3.4. FIM, CRLB and Numerical Search with Two Stages 96
3.5. Simulations 100
3.5.1. CRLB Evaluation Results 101
3.5.2. Results of Numerical Search with Two Stages 104
3.6. Summary and Conclusions 109

Ch. 4 : Heterogeneous Track-to-Track Fusion 114
4.1. Introduction 114
4.2. The Heterogenous Fusion Problem 116
4.3. Heterogenous Track-to-Track Fusion 118
4.3.1. The LMMSE Fuser 120
4.3.2. The ML Fuser 121
4.4. Crosscorrelation in Heterogeneous Fusion 122
4.5. Simulation Results 126
4.5.1. Active Sensor IMM Estimator Design 128
4.5.2. Passive Sensor KF Estimator Design 133
4.5.3. Results of Heterogeneous T2TF 135
4.6. Conclusions 139

Ch. 5 : Conclusions 143

App. A : Covariance Matrix in Unbiased Mixing 146

App. B : Unbiased Mixing for BLUE 147

App. C : Jacobian Matrix for Thrusting Mode 154

App. D : Taylor Expansion Approximation for LMMSE Fuser 157

App. E : Process Noise Covariance Matrix for NCT Model 159

App. F : MC Results for Sample Crosscorrelation 164

App. G : An Approximation Technique for Crosscovariance Matrix 165

App. H : Choice of Estimator for the Passive Sensor 172

App. I : Local Estimators' Consistency 174

Bibliography 2

LIST OF FIGURES

1.3.1 Cubic spline approximation of drag coef. multiplier 23
1.4.1 The MIMM approach. 26
1.6.1 A sample estimated trajectory (traj. 28). 35
1.6.2 Evolution of drag coef. estimates in MIMM (traj. 12) 39
1.6.3 Evolution of drag coef. estimates in MIMM (traj. 28) 40
1.6.4 Drag coef. and thrust estimates in the best IMM (traj. 12) 41
1.6.5 Drag coef. and thrust estimates in the best IMM (traj. 28) 42
1.6.6 Drag coef. and thrust estimates with conventional mixing (traj. 21) . 43
1.6.7 Drag coef. and thrust estimates with unbiased mixing (traj. 21) 44

2.2.1 The wind effect 50
2.3.1 IPP procedure based on the MIMM-UM estimator. 60
2.5.1 IPP uncertainty ellipse centered at true impact point 69
2.5.2 Clusters of MC IPP "clould of points" 74
2.5.3 IPP uncertainty ellipse and MC IPP cloud for various OTs 79
2.5.4 IPP uncertainty ellipse and MC IPP cloud for various accuracies . . . 80
2.5.5 IPP uncertainty ellipse and MC IPP cloud for various trajectories . . . 81
2.5.6 Consistency test (NEES in pos. and vel. and NIS, g.w.i.) 82

3.3.1 Ambiguity of 3-D trajectory using 2-D LoS measurements 94
3.5.1 Sample scenario with various platform behaviors. 102
3.5.2 CRLB-based SD as function of OT, known LP altitude 103
3.5.3 CRLB-based SD as function of OT for LP altitude parameter 104
3.5.4 CRLB-based SD as function of OT for various scenarios 105
3.5.5 CRLB-based SD as function of OT for various platform behaviors . . 111
3.5.6 Surface of inverse NLLF vs. launch angles 112
3.6.1 IPP procedure using numerical search with two stages 113

4.5.1 The scenario. 128
4.5.2 Probability of maneuvering mode (NCT) 137
4.5.3 Position RMSE for LMMSE fuser. 138
4.5.4 Velocity RMSE for LMMSE fuser. 139
4.5.5 NEES for LMMSE fuser. 140
4.5.6 Position RMSE for ML fuser. 141
4.5.7 Velocity RMSE for ML fuser. 142

F.0.1 Sample crosscorrelation for $\tilde{x}$ and $\tilde{y}$ with $\tilde{\theta}$ and $\dot{\tilde{\theta}}$. 165

F.0.2 Sample crosscorrelation for $\tilde{\dot{x}}$ and $\tilde{y}$ with $\tilde{\theta}$ and $\dot{\tilde{\theta}}$. 166
F.0.3 Sample crosscorrelation for $\tilde{\dot{x}}$ and $\tilde{y}$ with $\tilde{\theta}$ and $\dot{\tilde{\theta}}$. 167

F.0.4 Sample crosscorrelation for $\tilde{x}$ and $\tilde{y}$ with $\tilde{\theta}$ and $\tilde{\dot{\theta}}$. 168

H.0.5 Illustrated scenario for calculating target maneuvering index. 172

I.0.6 NEES for active sensor IMM and passive sensor KF. 175

LIST OF TABLES

1.3.1 Selected representative points for cubic spline interpolation 22
1.6.2 RMS change rate due to process noise (TM and BM) 34
1.6.3 IPP performance for 5 real trajectories (short range) 37
1.6.4 IPP performance for 15 real trajectories (medium range) 38
1.6.5 IPP performance for 10 real trajectories (long range) 38
1.6.6 IPP performance evaluation summary 42

2.4.7 Fitting of CV, CA and CJ models for trajectory 60C7H7 64
2.4.8 RMS change rate due to process noise (wind effect) 66
2.5.9 Parameters of the trajectories considered 68
2.5.10 Sensor accuracy settings 68
2.5.11 IPP RMS errors, caliber 60 mm 71
2.5.12 IPP RMS errors, caliber 81 mm 72
2.5.13 IPP RMS errors, caliber 120 mm 73

3.5.14 Search range and grid stepsize settings 106
3.5.15 Results of numerical search, known LP altitude 108
3.5.16 Results of numerical search, unknown LP altitude 109
3.5.17 Results of numerical search, 100 MC runs 109

4.5.18 RMS change rate due to process noise (NCV and NCT) 133

G.0.19 RMSE in position and velocity for LMMSE fuser 171

LIST OF ACRONYMS

BM	ballistic mode
BoP/BoT	burnout point/burnout time
CRLB	Cramér-Rao lower bound
CTF	centralized measurement tracker/fuser
DTAG	drag-coefficient-thrust-altitude grid
DTG	drag-coefficient-thrust grid
FIM	Fisher Information Matrix
HOT	higher order terms
HT2TF	heterogeneous track-to-track fusion
ILS	iterated least squares
IMM	interacting multiple model
IPP	impact point prediction
KF	Kalman filter
LF	likelihood function
LMMSE	linear minimum mean squared error
LoS	line of sight
LP	launch point
MC	Monte Carlo
MIMM	multiple interacting multiple model
ML	maximum likelihood
MP	moving platform
NCT	nearly coordinated turn
NCV	nearly constant velocity
N-D	N-dimensional
NDC	normalized drag coefficient
NEES	normalized estimation error squared
NIS	normalized innovation squared
NLLF	negative log-likelihood function
OT	observation time
PSD	power spectral density
RK	Runge-Kutta (4th order)
RMSE	root mean squared error
SD	standard deviation
SP	stationary platform
T2TF	track-to-track fusion
TM	thrusting mode
UM	unbiased mixing
e.w.i.	given wind information with a deterministic error
g.w.i.	given perfect knowledge of wind information
n.w.i.	given no knowledge of the wind information

Introduction

Two topics in the area of estimation, target tracking, data fusion are investigated: impact point prediction (IPP) for thrusting/ballistic projectiles and track-to-track fusion for heterogeneous sensors.

0.1 Motivation

0.1.1 State Estimation for IPP with an Active Sensor

The problem of estimating the state of thrusting/ballistic endoatmospheric projectiles for the end purpose of IPP has attracted wide attention. The trajectory of a thrusting projectile can be divided into two phases: thrusting and ballistic. The target dynamics in different phases are substantially different and characterized by different main forces. The resulting acceleration consists of the thrust, drag, and gravity components [12] (the lift force is neglected in this work, since it is very small in the scenarios considered). For the thrusting phase, we consider thrust, drag, and gravity; for the ballistic phase, only gravity and drag.

Given observations from an active sensor over a very short time, we face two major challenges for the IPP problem. First, different phases of the trajectory

require different mathematical models to describe the corresponding physical behaviors — there is uncertainty as to which motion model is in effect. Secondly, in practical situations, for a trajectory with unknown drag coefficient and unknown thrust, it is difficult to separate between them given a very short observation time — there is an ambiguity in drag coefficient-thrust[1] estimation. The uncertainty in the motion model and ambiguity in the drug-thrust estimation are acting simultaneously. This significantly affects the results of the state estimation (including the drag coefficient and thrust components) and the IPP that follows. It is worth noting that the IPP accuracy depends mainly on the drag coefficient estimate since the prediction is done after the thrusting period is over. A good initial estimate of the drag coefficient is conducive to a more accurate estimate of the thrust, which then results in a more accurate overall state estimate and better IPP performance.

We propose a multiple interacting multiple model (MIMM) estimator using different initial drag coefficients to alleviate the model uncertainty and estimation ambiguity. For each IMM from the MIMM estimator, a thrust mode (TM) and a ballistic mode (BM) are used to match the thrusting phase and ballistic phase. Since the TM and BM state vectors are of unequal dimensions, an unbiased IMM mixing procedure — useful for quite general applications — is presented. The IMM with unbiased mixing (IMM-UM) provides unbiased estimates for the extra component(s), i.e., the thrust in the present case. The

[1]For simplicity, "drag coefficient-thrust" will be called "drag-thrust" if there is no ambiguity.

sensitivity of the system performance to the drag coefficient estimate leads to using a multiple IMM (MIMM) estimator to overcome the drag-thrust ambiguity. The technique involves evaluating the likelihood functions of the IMM estimators with different initial drag coefficient estimates to quantify how well the multiple IMM estimators fit the observation data. Compared to the single IMM estimator with conventional mixing, the proposed MIMM-UM approach is shown to provide improved IPP performance in practical applications.

The IPP problem is further extended under a more realistic and complicated environment, namely, in the presence of wind. The wind effect exacerbates the estimation ambiguity between drag and thrust and induces additional uncertainty in the IPP procedure. A tracker with its dynamic model modified (from the afore-mentioned case in the absence of wind) to incorporate the wind effect is presented. An extensive analysis of the wind effect on the IPP system using the MIMM-UM estimator is conducted for various wind scenarios by Monte Carlo (MC) simulations.

0.1.2 Parameter Estimation for IPP with a Passive Sensor

The estimation of the state of thrusting/ballistic endoatmospheric projectiles moving in 3-dimensional (3-D) space using 2-dimensional (2-D) measurements from a single passive sensor for the end purpose of IPP is a more challenging problem. Besides the difficulties of the IPP problem using full position measurements from an active sensor, two major issues must be dealt with. The first

is that to estimate and predict the 3-D trajectory using the (incomplete) 2-D line of sight (LoS, i.e., azimuth and elevation angles) measurements, instead of full position measurements [8][34], could exhibit a geometry-dependent ambiguity due to incomplete observability. This requires an estimability analysis for the target parameters that uniquely determine the trajectory. The second is that lack of knowledge about the location of the projectile's launch point (LP) exacerbates the estimation ambiguity between the drag coefficient and thrust estimands [34].

Estimability analysis for such a nonlinear system is a challenging task. In the literature the Fisher information matrix (FIM) has been used as a convenient tool in target motion analysis problems [10][26] to analyze estimability. The major difficulty of obtaining the estimability criterion lies in the fact that the FIM can only be obtained numerically rather than analytically due to the nonlinear motion equations which require numerical integration. The corresponding Cramér-Rao lower bound (CRLB) provides (if the estimator is efficient) the accuracy of the parameter estimate, which can be mapped into IPP accuracy. Motion of the passive sensor could enhance the system estimability. However, due to the unavoidable numerical integration, the improvement of a moving platform (MP) over the corresponding stationary platform (SP) could not be shown analytically.

The availability of the LP position information is very crucial for the system performance. An accurate estimate for the LP will lead to the target parame-

ter estimate being mapped into the IPP of good accuracy. In many practical situations, the full LP position is generally unavailable but the altitude of the LP can be available from terrain information. The launch position is then obtained from the first LoS measurement intersected with the terrain elevation database. If the terrain elevation (altitude) information is unavailable, the estimation problem becomes substantially more difficult. The LP altitude is then an additional unknown target parameter, used to specify the unknown LP location information.

In this study, the burnout time (BoT) is assumed available from the passive (optical) sensor and we make two assumptions about the important LP location information: known LP altitude and unknown LP altitude. The estimability of the target parameter vector that uniquely determines its trajectory is analyzed by checking the invertibility of the FIM. A maximum likelihood (ML) estimator for the target parameters is developed. Once the FIM is found to be full-rank, the ML estimate can be obtained by seeking the global maximum of the likelihood function (LF) of the target parameters and the corresponding CRLB can be used for the IPP accuracy evaluation. With the aid of the CRLB, the motion of the passive sensor to enhance the system performance is investigated. A search strategy with two stages — a mixed (partially grid-based) search followed by a continuous search — is proposed. Due to its parallelizable nature, the mixed search allows the two-stage strategy to be real-time implementable.

0.1.3 Heterogeneous Track-to-Track Fusion

In linear multisensor tracking systems the best target state estimation performance is obtained by a centralized tracker/fuser (CTF), by directly sending to the fusion center (FC) all the measurements of the local sensors. The CTF is also known as measurement-to-track fuser and its superiority over other information configurations can be proved only for the linear case [3]. In many practical situations, because of communication constraints, each local sensor has its own information processing system and sends only tracks to the FC, which fuses appropriately tracks from different local sensors to achieve comparable estimation performance to that of the CTF [3].

Track-to-track fusion using estimates from multiple sensors can achieve better estimation performance than single sensor tracking. If the local sensors use different system models in different state spaces, the problem of heterogeneous track-to-track fusion arises. Compared with homogeneous track-to-track fusion that assumes the same system model for different sensors, the heterogeneous case poses two major challenges. The first one is that we have to fuse estimates from different state spaces (related by a certain nonlinear transformation). The second is the estimation errors' dependence problem due to the "common process noise effect" [3] and there is no known way to capture the common" part exactly.

In the literature there are few works dealing with the model heterogeneity. A heterogeneous T2TF (HT2TF) approach was presented in [6] to fuse the tracks from an active sensor and a passive sensor with different state vectors.

However, the fusion was done by using the full Cartesian state estimates (from an active sensor) to update the smaller angular state estimates (from a passive sensor).

The major difficulty to evaluate the dependence of the estimation errors due to the common process noise effect lies in how to capture the "common" part of process noises from different state spaces. The dependence of the estimation errors can be quantified by the crosscovariance matrix, and the more accurately the crosscovariance matrix is obtained, the better the heterogeneous track-to-track fusion performance will be. However, the difference between the motion models for different sensors prohibits the evaluation of the crosscovariance matrix by the exact method described in [3] (which is limited to the homogeneous case and linear systems).

In this study, we propose a HT2TF by using the smaller vector of angular state estimates (from a passive sensor) to update the full Cartesian state estimates (from an active sensor). Unlike in the homogeneous T2TF, we observe indefinite (sometimes positive and sometimes negative) crosscorrelation coefficients in the heterogenous case. In a study case using an IMM tracker, the HT2TF (a nonlinear fuser) allows each local sensor to flexibly design a more suitable local estimator which was shown to lead to a better estimation performance than the corresponding CTF, unlike in the (linear) homogeneous case.

0.2 Outline

Chapter 1 describes the multiple-IMM estimation approach with unbiased mixing for the IPP of thrusting projectiles (in the absence of wind). Section 1.2 introduces the novel unbiased mixing approach to deal with mixing the state estimates of modes with unequal dimensions. Section 1.3 presents the system model and the corresponding discretized form. The IMM likelihood function used in selecting the best IMM estimator from the MIMM estimator is presented in Section 1.4 and an impact point prediction (IPP) procedure is described in Section 1.5. In Section 1.6, the IMM estimator design and the estimation and IPP results for 30 real-data scenarios are given. Conclusions are presented in Section 1.7.

Chapter 2 describes the IPP problem using the MIMM-UM estimator in the presence of wind. In Section 2.2, the wind effect and the modified dynamic model in the presence of wind are presented. Section 2.3 outlines the MIMM-UM estimator and the IPP procedure. Section 2.4 presents the design of the MIMM estimator and a N-point adaptive initialization approach. Then the IPP performance is investigated in the presence of different wind (strength and direction) conditions for various total observation time and sensor accuracy scenarios by simulations in Section 2.5. Conclusions are presented in Section 2.6.

Chapter 3 presents a parameter estimation approach for IPP in 3-D using the 2-D measurements from a single passive sensor. Section 3.2 formulates the problem and defines the notations. In Section 3.3, the possible existence of a

geometry-dependent ambiguity due to incomplete observability is discussed and the ML estimator of the target parameter vector is developed. Section 3.4 presents, under the assumptions of known and unknown LP altitude, the FIM, the CRLB and a search strategy with two stages for the MLE. In Section 3.5, the simulation scenarios and results are presented. Section 3.6 provides the conclusions.

Chapter 4 presents the track fusion for heterogeneous sensors. Section 4.2 formulates the heterogenous T2TF problem. Two approaches, namely, the linear minimum mean square error (LMMSE) and maximum likelihood (ML) heterogeneous T2TF are presented in Section 4.3. The crosscorrelation analysis by MC simulations is presented in Section 4.4. Section 4.5 evaluates the proposed approaches in a tracking scenario with an active sensor and a passive sensor. Section 4.6 provides the conclusions.

0.3 Publications

The research towards this dissertation resulted in the following publications.

Journals:

1. T. Yuan, Y. Bar-Shalom and X. Tian, "Heterogeneous Track-to-Track Fusion", *J. Advances in Information Fusion.* 6(2): 131-149, Dec. 2011.
2. T. Yuan, Y. Bar-Shalom, P. K. Willett, E. Mozeson, S. Pollak and D. Hardiman, "A Multiple IMM Estimation Approach with Unbiased Mixing for Thrusting Projectiles", *IEEE Trans. Aerosp. Electronic Systems*, 48(4):3250–

3267, Oct. 2012.

3. T. Yuan, Y. Bar-Shalom, P. K. Willett and D. Hardiman, "Impact Point Prediction for Thrusting Projectiles in the Presence of Wind", *IEEE Trans. Aerosp. Electronic Systems*, to appear, 2013.
4. T. Yuan, Y. Bar-Shalom, P. K. Willett, R. Ben-Dov and S. Pollak "Estimation of Thrusting Trajectories in 3-D from a Single Fixed Passive Sensor", *IEEE Trans. Aerosp. Electronic Systems*, Accepted, 2013.

Conference Proceedings:

1. T. Yuan, Y. Bar-Shalom and X. Tian, "Heterogeneous Track-to-Track Fusion", *Proc. of the 14th International Conference on Information Fusion*, pp.1-8, Chicago, IL, July 2011.
2. T. Yuan, Y. Bar-Shalom, P.K. Willett and D. Hardiman, "Impact Point Prediction for Short Range Thrusting Projectiles", *Proc. SPIE Conf. Signal and Data Processing of Small Targets*, #7698-55, Orlando, FL, April 2010.
3. T. Yuan, Y. Bar-Shalom, P. K. Willett and D. Hardiman, "Impact Point Prediction for Thrusting Projectiles in the Presence of Wind", *Proc. SPIE Conf. Signal Processing, Sensor Fusion, and Target Recognition XXI*, #8392-63, Baltimore, MD, April 2012.
4. T. Yuan, Y. Bar-Shalom, P. K. Willett and R. Ben-Dov and S. Pollak, "Estimability of Thrusting Trajectories in 3-D from a Single Passive Sensor", *Proc. SPIE Conf. Signal Processing, Sensor Fusion, and Target Recognition XXII*, #8745-01, Baltimore, MD, April 2013.

5. T. Yuan, Y. Bar-Shalom, P. K. Willett and R. Ben-Dov and S. Pollak, "Estimability of Thrusting Trajectories in 3-D from a Single Passive Sensor with Unknown Launch Point", *Proc. SPIE Conf. Signal and Data Processing of Small Targets,* #8857-12, San Diego, CA, Aug. 2013.

Chapter 1

A Multiple IMM Estimation Approach with Unbiased Mixing for IPP

We present a procedure to estimate the state of thrusting/ballistic endoatmospheric projectiles for the end purpose of impact point prediction. The short observation time and the estimation ambiguity between drag coefficient and thrust in the dynamic model motivate the development of a multiple interacting multiple model (MIMM) estimator with various drag coefficient initializations. A simple unbiased IMM mixing procedure (useful for quite general applications) is presented for state estimators with unequal dimensions and applied for the thrusting and ballistic modes in the case considered. Results with real data are given.

1.1 Introduction

The trajectory of a thrusting projectile can be divided into two phases: thrusting and ballistic. The target dynamics are substantially different and characterized by different main forces in each phase. The resulting acceleration can be written in terms of the thrust, drag, and gravity components [12] (the lift force is

neglected in this work, since it is very small in the scenarios considered) as

$$\mathbf{a}_F = \mathbf{a}_T + \mathbf{a}_D + \mathbf{a}_G \tag{1.1.1}$$

For the thrusting phase, we consider thrust, drag, and gravity; for the ballistic phase, only drag and gravity. For a trajectory with unknown drag coefficient and unknown thrust, it is difficult to clearly divide the trajectory into its phases. The interacting multiple model (IMM) estimator with a thrust mode (TM) and a ballistic mode (BM) to match these two phases is a natural choice (it is assumed that no optical indication of the plume is available, as in the real-data examples considered).

Given a very short observation period of a thrusting/ballistic projectile for the purpose of impact point prediction (IPP), an accurate estimate of the target state (including its thrust and drag coefficient) is of the utmost importance. The IPP accuracy depends mainly on the drag coefficient estimate since the prediction is done after the thrusting period is over, i.e., based on the BM. Note that the net acceleration (1.1.1) includes the (algebraic) sum of the drag and thrust, thus causing an ambiguity in the estimation when both are unknown. A better initial estimate of the drag coefficient is conducive to a more accurate estimate of the thrust, which then results in a more accurate overall state estimate and better IPP performance. The sensitivity of the system performance to the drag coefficient estimate leads to using a multiple-IMM (MIMM) estimator

to overcome the above-mentioned ambiguity to more accurately estimate the drag coefficient and thrust. The technique involves evaluating the likelihood functions of the IMM estimators with different initial drag coefficient estimates to quantify how well the multiple IMM estimators fit the observation data [19].

Another issue that arises is the mixing of the mode-conditioned state estimates in an IMM estimator when the modes used have different dimension state vectors. The conventional approach in this case is to augment with zeros the lower dimension state estimate prior to the mixing [3]. However, this leads to a bias toward zero for the state components of the larger state vector that are mixed with the extra components of the smaller state that are zero. A simple procedure to avoid this "bias" is presented, together with a suitable augmentation of the covariance of the smaller state that yields an unbiased and consistent mixing. This leads to a "direct" block mixing in contrast to the component by component mixing described in the recent work [13][2].

Compared to a purely ballistic projectile [21] the IPP for thrusting projectiles is a significantly more challenging problem. When tracking a thrusting projectile, there is an estimation ambiguity between drag and thrust in the dynamic model: a sudden decrease in the thrust estimate causes a sudden increase in the drag coefficient estimate and vice versa. The unbiased mixing avoids the mixing of the thrust estimate from the thrusting mode filter with zero from

[2]The unbiased mixing work detailed here dates back to [31]. Our work on unbiased mixing and the work of [13] were carried out independently, with the latter considering a more general formulation.

the ballistic mode filter (the latter assumes no thrust), which would bias the mixed thrust estimate towards zero. This is the major difference between [21] and the present work. The MIMM approach gives the best choice of initial drag coefficient estimate, based on the characteristics of the thrusting projectile trajectory observations.

The chapter is organized in following manner. We first give a brief overview of the standard IMM estimator in Section 1.2 and the novel unbiased mixing approach is introduced here to deal with mixing the state estimates of modes with different dimensions. The dynamic and the measurement model and the corresponding discretized form are presented in Section 1.3. The IMM likelihood function used in selecting the best initial drag coefficient estimate from the MIMM estimator is presented in Section 1.4. The IPP procedure is described in Section 1.5. The IMM estimator design and the estimation and prediction results for 30 real-data scenarios are given in Section 1.6. Conclusions are presented in Section 1.7.

1.2 IMM Estimator with Unbiased Mixing

In order to estimate the target state during the short observation interval as accurately as possible and then carry out the IPP, we use an approach that relies on the IMM estimator with two different mode-matched filters for the thrust mode (TM) and the ballistic mode (BM).

Generally, in an IMM estimator with r modes that run in parallel, the state

estimates at the beginning of a cycle are a mixture of estimates of the previous cycle based on all the modes using the "mixing" probabilities [2]. In the conventional IMM estimator, mixing the estimates from different modes with unequal dimension state vectors causes a bias for the *extra* components of the higher dimension state vector [3]. It is necessary to introduce an unbiased approach for the mixing, especially when the extra components contain information of special interest, as in the case of high-thrust projectiles.

Using the notations from [2] and omitting the time index for simplicity, we assume an IMM estimator with two modes, say, mode 1 and mode 2. More general cases can be handled in a similar manner. The state of mode 1 has a higher dimension

$$\mathbf{x}^1 \triangleq \begin{bmatrix} \mathbf{x}_c \\ \\ \mathbf{x}_e \end{bmatrix} \tag{1.2.1}$$

and the state of mode 2, which has lower dimension (it consists of a subset of the components of $\mathbf{x}^1$), can be augmented as

$$\mathbf{x}^2 \triangleq \begin{bmatrix} \mathbf{x}_c \\ \\ 0 \end{bmatrix} \tag{1.2.2}$$

where $\mathbf{x}_c$ denotes the common ('c') components of states of the two modes and $\mathbf{x}_e$ consists of the extra ('e') state components of mode 1 compared with that of mode 2.

The estimates of modes 1 and 2 are denoted as

$$\hat{\mathbf{x}}^1 \triangleq \begin{bmatrix} \hat{\mathbf{x}}_c \\ \\ \hat{\mathbf{x}}_e \end{bmatrix} \tag{1.2.3}$$

with the corresponding covariance matrix

$$P^1 \triangleq \begin{bmatrix} P_c^1 P_{ce}^1 \\ \\ P_{ec}^1 P_e^1 \end{bmatrix} \tag{1.2.4}$$

and

$$\hat{\mathbf{x}}^2 \triangleq \begin{bmatrix} \hat{\mathbf{x}}_c^2 \\ \\ 0 \end{bmatrix} \tag{1.2.5}$$

with the corresponding covariance matrix

$$P^2 \triangleq \begin{bmatrix} P_c^2 0 \\ \\ 0\ 0 \end{bmatrix} \tag{1.2.6}$$

respectively, where, according to the conventional procedure [3], the lower dimension state is augmented with zeros to make it of the same dimension as the higher dimension state.

We denote the mixing probabilities [2] as

$$\mu_{i|j} \triangleq P\left\{O^i(k-1)|O^j(k), Z^{k-1}\right\} \qquad i, j \in \{1, 2\} \tag{1.2.7}$$

where $O^d(k)$ is the possible mode of the mode-matched filter d at time k.

The standard mixing of the two estimates yields

$$\hat{\mathbf{x}}^{01} \triangleq \begin{bmatrix} \hat{\mathbf{x}}_c^{01} \\ \hat{\mathbf{x}}_e^{01} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}_c^1 \\ \hat{\mathbf{x}}_e^1 \end{bmatrix} \mu_{1|1} + \begin{bmatrix} \hat{\mathbf{x}}_c^2 \\ 0 \end{bmatrix} \mu_{2|1} = \begin{bmatrix} \hat{\mathbf{x}}_c^1 \mu_{1|1} + \hat{\mathbf{x}}_c^2 \mu_{2|1} \\ \hat{\mathbf{x}}_e^1 \mu_{1|1} \end{bmatrix} \tag{1.2.8}$$

and

$$\hat{\mathbf{x}}^{02} \triangleq \begin{bmatrix} \hat{\mathbf{x}}_c^{02} \\ \hat{\mathbf{x}}_e^{02} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}_c^1 \\ \hat{\mathbf{x}}_e^1 \end{bmatrix} \mu_{1|2} + \begin{bmatrix} \hat{\mathbf{x}}_c^2 \\ 0 \end{bmatrix} \mu_{2|2} = \begin{bmatrix} \hat{\mathbf{x}}_c^1 \mu_{1|2} + \hat{\mathbf{x}}_c^2 \mu_{2|2} \\ \hat{\mathbf{x}}_e^1 \mu_{1|2} \end{bmatrix} \tag{1.2.9}$$

Note that the last component of (1.2.8) is multiplied by a factor $\mu_{1|1}$, less than one, while the last component of (1.2.9) is ignored by the lower dimension (mode 2) filter. Thus, there is an estimate bias for the extra component(s) in the filter matched to mode 1 at the start of each IMM cycle.

The bias is eliminated by replacing (1.2.5) in the mixing (1.2.8) according to the following modification ('M')

$$\hat{\mathbf{x}}_M^2 \triangleq \begin{bmatrix} \hat{\mathbf{x}}_c^2 \\ \hat{\mathbf{x}}_e^1 \end{bmatrix} \tag{1.2.10}$$

as if mode 2 has the same extra component(s) estimate (thrust in our case) as mode 1.

Then the modified estimate mixing replaces (1.2.8) by

$$\hat{\mathbf{x}}_M^{01} \triangleq \hat{\mathbf{x}}^1 \mu_{1|1} + \hat{\mathbf{x}}_M^2 \mu_{2|1} = \begin{bmatrix} \hat{\mathbf{x}}_c^1 \mu_{1|1} + \hat{\mathbf{x}}_c^2 \mu_{2|1} \\ \hat{\mathbf{x}}_e^1 \end{bmatrix} \triangleq \begin{bmatrix} \hat{\mathbf{x}}_c^{01} \\ \hat{\mathbf{x}}_{Me}^{01} \end{bmatrix} \tag{1.2.11}$$

i.e., the extra component(s) $\hat{\mathbf{x}}_{Me}^{01}$ of $\hat{\mathbf{x}}_{M}^{01}$ are exactly the same as those of $\hat{\mathbf{x}}^{1}$ and, consequently, there is no bias after the modified mixing.

Next, the modification from (1.2.8) to (1.2.10) should also be accounted for in the covariance resulting from the mixing. The corresponding covariance matrix for the modified mixing will be (the details are shown in Appendix A)

$$P_M^{01}=\left\{\begin{bmatrix} P_c^1 & P_{ce}^1 \\ P_{ec}^1 & P_e^1 \end{bmatrix}+\begin{bmatrix} \tilde{P}_c^1 & \tilde{P}_{Mce}^1 \\ \tilde{P}_{Mec}^1 & \tilde{P}_{Me}^1 \end{bmatrix}\right\}\mu_{1|1}+\left\{\begin{bmatrix} P_c^2 & 0 \\ 0 & P_e^1 \end{bmatrix}+\begin{bmatrix} \tilde{P}_c^2 & 0 \\ 0 & 0 \end{bmatrix}\right\}\mu_{2|1} \tag{1.2.12}$$

The blocks of P_M^{01} are obtained as

$$P_{Mc}^{01}\overset{\Delta}{=}\left(P_c^1+\tilde{P}_c^1\right)\mu_{1|1}+\left(P_c^2+\tilde{P}_c^2\right)\mu_{2|1} \tag{1.2.13}$$

$$P_{Me}^{01}\overset{\Delta}{=}P_e^1\mu_{1|1}+P_e^1\mu_{2|1}=P_e^1 \tag{1.2.14}$$

$$P_{Mce}^{01}\overset{\Delta}{=}(P_{ce}^1+\tilde{P}_{Mce}^1)\mu_{1|1} \tag{1.2.15}$$

where, according to equation (1.4.16-9) in [2], the spread-of-the-means terms are

$$\tilde{P}_c^1=(\hat{\mathbf{x}}_c^1-\hat{\mathbf{x}}_c^{01})(\hat{\mathbf{x}}_c^1-\hat{\mathbf{x}}_c^{01})' \tag{1.2.16}$$

$$\tilde{P}_c^2=(\hat{\mathbf{x}}_c^2-\hat{\mathbf{x}}_c^{01})(\hat{\mathbf{x}}_c^2-\hat{\mathbf{x}}_c^{01})' \tag{1.2.17}$$

Note that, in view of (1.2.11), one has

$$\tilde{P}_{Mce}^1\overset{\Delta}{=}(\hat{\mathbf{x}}_c^1-\hat{\mathbf{x}}_c^{01})(\hat{\mathbf{x}}_e^1-\hat{\mathbf{x}}_{Me}^{01})'=0 \quad (\tilde{P}_{Mec}^1=0) \tag{1.2.18}$$

$$\tilde{P}_{Me}^1\overset{\Delta}{=}(\hat{\mathbf{x}}_e^1-\hat{\mathbf{x}}_{Me}^{01})(\hat{\mathbf{x}}_e^1-\hat{\mathbf{x}}_{Me}^{01})'=0 \tag{1.2.19}$$

The mixed estimate $\hat{\mathbf{x}}_c^{02}$ and its covariance matrix P_c^{02} are obtained in the standard manner, using $\hat{\mathbf{x}}_c^1$, P_c^1, $\hat{\mathbf{x}}_c^2$ and P_c^2 with the corresponding mixing probabilities.

Note that the common components after the unbiased mixing have the same covariance matrix as that in the conventional mixing but the extra component (thrust) variance stays unchanged after the unbiased mixing.

A generalized form for the unbiased mixing based on the best linear unbiased estimator (BLUE) is discussed in Appendix B.

1.3 System Model and its Discretized Form

For the problem considered, the higher dimension state, i.e., the state vector for the thrusting mode, is denoted as

$$\mathbf{x}(t) = [\, x(t)\ y(t)\ z(t)\ \dot{x}(t)\ \dot{y}(t)\ \dot{z}(t)\ \alpha(t)\ \tau(t)\,]' \tag{1.3.1}$$

where $\alpha(t)$ is the drag coefficient and $\tau(t)$ is the thrust. The time arguments will be omitted where there is no ambiguity.

It is known that the drag coefficient varies significantly with the Mach number regime: subsonic, transonic and supersonic. This will be accounted for by a Mach number-dependent multiplier.

The dynamic model for this problem can be written as follows

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{\tau}{V} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} + \alpha\, \alpha_m\, D \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} + g \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} + \tilde{v}_1 \tag{1.3.2}$$

and

$$\dot{\alpha} = \tilde{v}_2 \tag{1.3.3}$$

$$\dot{\tau} = \tilde{v}_3 \tag{1.3.4}$$

where

- The first term on the right side of (1.3.2) is the thrust in the x, y, and z directions. For the ballistic phase, the thrust is zero.
- The second term is the drag part, which is related to target velocity and altitude.
- V is the magnitude of the velocity $v = [\,\dot{x}\ \ \dot{y}\ \ \dot{z}\,]'$, i.e., the speed (in m/ s).
- α is the drag coefficient (in m^2/kg) at subsonic speed and τ is the thrust (in $\mathrm{m/s}^2$).
- α_m is the (dimensionless) *Mach number-dependent drag coefficient multiplier*, which approximated by a cubic spline curve shown in Fig. 1.3.1. This is obtained by interpolating the selected representative points shows in Table 1.3.1 (speed vs. normalized drag coefficient — NDC).
- $D = -\frac{\rho(z)V}{2}$, where $\rho(z) = \rho_0 e^{-cz}$ is the air density (in $\mathrm{kg/m}^3$) at altitude z

(in m) and c is the air density constant (in m^{-1}) [21].

- g is the standard acceleration due to gravity at sea level, assumed to be the same throughout the trajectory, with value 9.812 m/s^2.
- $\tilde{v}_1$, $\tilde{v}_2$, and $\tilde{v}_3$ are assumed to be continuous-time zero-mean white Gaussian noises. The drag coefficient and thrust acceleration are thus modeled as Wiener processes with slow variation [2].

Table 1.3.1: Selected representative points for cubic spline interpolation

Speed	1	50	100	150	190	230	280	285	295	310	350
NDC	1	1.01	1.015	1.02	1.03	1.035	1.1	1.2	1.3	1.4	1.59
Speed	400	440	500	550	590	650	750	800	850	1000	1000 +
NDC	1.69	1.71	1.68	1.55	1.4	1.35	1.33	1.32	1.31	1.3	1.3

Combining the dynamic equations (1.3.2)–(1.3.4), we have the following compact form

$$\dot{\mathbf{x}}(t) = f[\mathbf{x}(t)] + \tilde{v}(t) \tag{1.3.5}$$

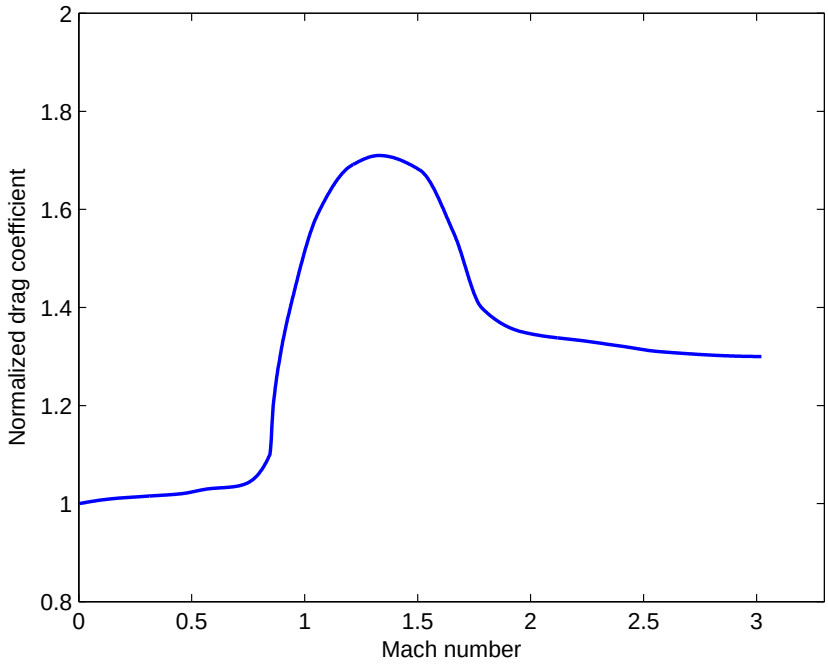


Figure 1.3.1: Cubic spline approximation of the Mach number-dependent drag coefficient multiplier (for a "sharp nose" projectile [36]).

where

$$f(\mathbf{x}(t)) = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \\ \tau(t)\dfrac{\dot{x}(t)}{V(t)} + \alpha(t)\alpha_m(t)D(t)\dot{x}(t) \\ \tau(t)\dfrac{\dot{y}(t)}{V(t)} + \alpha(t)\alpha_m(t)D(t)\dot{y}(t) \\ \tau(t)\dfrac{\dot{z}(t)}{V(t)} + \alpha(t)\alpha_m(t)D(t)\dot{z}(t) - g \\ 0 \\ 0 \end{bmatrix} \tag{1.3.6}$$

and

$$\tilde{v}(t) = [\, \tilde{v}_1(t)' \;\; \tilde{v}_2(t) \;\; \tilde{v}_3(t) \,]' \tag{1.3.7}$$

The state vector equation is discretized by a second order Taylor expansion [15] as

$$\mathbf{x}(t+T) = \mathbf{x}(t) + \dot{\mathbf{x}}(t)T + \ddot{\mathbf{x}}(t)\frac{T^2}{2} + \text{HOT} \tag{1.3.8}$$

where T is the sampling time interval and HOT denotes the higher order terms that are neglected. Note that

$$\ddot{\mathbf{x}}(t) = \frac{d\dot{\mathbf{x}}(t)}{dt} = \frac{df[\mathbf{x}(t)]}{dt} = \frac{\partial f[\mathbf{x}(t)]}{\partial \mathbf{x}}\dot{\mathbf{x}}(t) \tag{1.3.9}$$

Using the discrete time notation $\mathbf{x}(k) = \mathbf{x}(t)|_{t=kT}$ and $\mathbf{x}(k+1) = \mathbf{x}(t+T)|_{t=kT}$, we have the following expression for (1.3.8) with the discretized continuous-time white process noise

$$\mathbf{x}(k+1) = \mathbf{x}(k) + f[\mathbf{x}(k)]T + A(k)f[\mathbf{x}(k)]\frac{T^2}{2} + \nu(k) \tag{1.3.10}$$

where $A(k)$ is the Jacobian of (1.3.6) evaluated at $\mathbf{x}(k)$ and $\nu(k)$ is the discretized continuous time process noise for the sampling interval T. Based on the assumption that α is nearly constant and D is related to both z and V, the detailed form of A is given in the Appendix C.

The corresponding covariance matrix of the discretized process noise is

$$Q = \begin{bmatrix} \begin{bmatrix} \frac{T^3}{3} I_3 & \frac{T^2}{2} I_3 \\ \frac{T^2}{2} I_3 & T I_3 \end{bmatrix} q_v & \mathbf{0}_{6\times 1} & \mathbf{0}_{6\times 1} \\ \mathbf{0}_{1\times 6} & T q_\alpha & 0 \\ \mathbf{0}_{1\times 6} & 0 & T q_\tau \end{bmatrix} \tag{1.3.11}$$

where I_3 is the 3×3 identity matrix and the continuous time process noise "intensities" q_v, q_α and q_τ are the corresponding power spectral densities (PSD).

The measurement model in Cartesian coordinates is

$$\mathbf{z}(k) = \begin{bmatrix} x(k) + w_1(k) \\ y(k) + w_2(k) \\ z(k) + w_3(k) \end{bmatrix} \stackrel{\Delta}{=} H\mathbf{x}(k) + \mathbf{w}(k) \tag{1.3.12}$$

where $H = [\, I_3 \;\; \mathbf{0} \,]$ and the components of the measurement noise vector $\mathbf{w}(k)$ are independent zero mean white Gaussian noises with the same standard deviation (SD) σ_p.

Assuming the position measurements are obtained from intersection of LoS (line of sight) observations from two or more cameras (i.e., configuration III multisensor data fusion [3]), the procedure of [18] can be used to obtain the exact position measurement noise covariance matrix. This requires the relative sensor-target geometry and the sensor accuracies (not available in the real-data examples considered in our work).

1.4 Best Initial Drag Coefficient Estimate of MIMM Estimator

In the TM dynamic model, the drag parameter (drag coefficient) and thrust parameter are separate state components as shown in (1.3.1). However, the drag force and thrust force are acting simultaneously (see equation (1.1.1)) and the IMM estimator has difficulty distinguishing between them if the initial uncertainty in the drag coefficient is large. A sudden decrease in the drag coefficient estimate may trigger an increase in the thrust estimate. The sensitivity of the estimation to the initial drag coefficient estimate necessitates the use of an MIMM estimator to overcome this "marginal observability" problem [2].

The procedure starts by establishing a set of L IMM estimators, each with an appropriate set of modes (TM and BM in our case) to describe the system behavior. Each IMM estimator will be initialized with a different value of the drag coefficient with a suitable initial standard deviation (SD) (generally, the initial SD is taken equal to 25% of the initial estimate of the drag coefficient). The filtering parameters are discussed in Section 1.6.

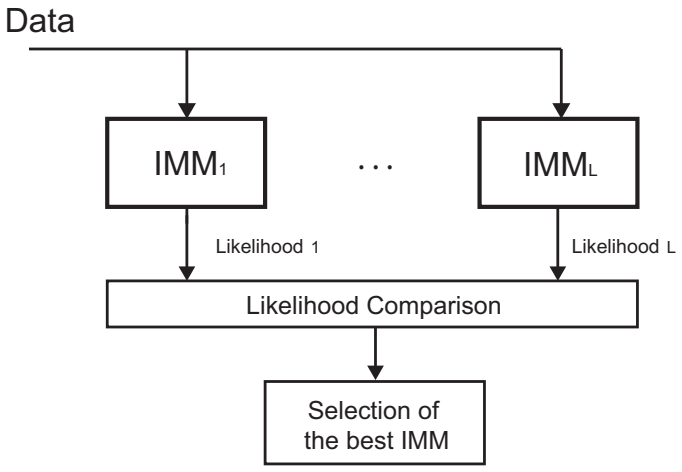


Figure 1.4.1: The MIMM approach.

In order to select the best initial drag coefficient estimate, we need to determine the most likely one of the L IMM estimators, during the observation period (see Fig. 1.4.1). The likelihood function of IMM estimator l for the time interval $[k_0, K]$ is

$$\Lambda_l^{k_0,K} = \prod_{k=k_0}^{K} \Lambda_l(k) \qquad l = 1, \ldots, L \tag{1.4.1}$$

where $\Lambda_l(k)$ is the likelihood function of lth IMM estimator, with r modes, at time k, given by [3]

$$\Lambda_l(k) \stackrel{\Delta}{=} \sum_{i=1}^{r} \Lambda_l^i(k) \mu_l^i(k|k-1) \tag{1.4.2}$$

where $\Lambda_l^i(k)$ is the likelihood function of mode i of IMM estimator l at time k and $\mu_l^i(k|k-1)$ is the predicted mode probability for mode i of IMM estimator l. The mode likelihood is [2]

$$\begin{aligned} \Lambda_l^i(k) &= p\left[\mathbf{z}(k)|M_l^i(k), \hat{\mathbf{x}}_l^{0i}(k-1|k-1), P_l^{0i}(k-1|k-1)\right] \\ &= \mathcal{N}\left[\mathbf{z}(k); \hat{\mathbf{z}}_l^i(k|k-1), S_l^i(k)\right] \end{aligned} \tag{1.4.3}$$

where $\mathcal{N}[\cdot]$ is the Gaussian probability density function, $\hat{\mathbf{z}}_l^i$ is the predicted measurement and S_l^i is the innovation covariance in mode i of IMM estimator

l. The predicted mode probability can be written as

$$\begin{aligned}\mu_l^i(k|k-1) &= \sum_{j=1}^{r} P\left\{M_l^i(k)|M_l^j(k-1), Z^{k-1}\right\} P\left\{M_l^j(k-1)|Z^{k-1}\right\} \\ &\triangleq \sum_{j=1}^{r} p_{ji}\mu_l^j(k-1) \end{aligned} \tag{1.4.4}$$

where $\mu_l^j(k-1)$ is the mode probability of mode j of IMM estimator l at time $k-1$ and p_{ji} is the transition probability from mode j to mode i over one time interval.

Using (1.4.3) and (1.4.4), one can rewrite (1.4.2) as

$$\Lambda_l(k) = \sum_{j=1}^{r} \left\{ \mathcal{N}\left[\mathbf{z}(k); \hat{\mathbf{z}}_l^i(k|k-1), S_l^i(k)\right] \sum_{j=1}^{r} p_{ji}\mu_l^j(k-1) \right\} \tag{1.4.5}$$

Then

$$l^* = \arg\max_l \Lambda_l^{k_0,K} = \arg\min_l \left[-\ln \Lambda_l^{k_0,K}\right] \tag{1.4.6}$$

selects the *best IMM estimator*. This yields the best initial estimate of the drag coefficient.

1.5 Impact Point Prediction

Based on the MIMM estimator, we can choose at the end of observation period the most likely initial drag coefficient of the most probable mode from the best IMM estimator (1.4.6). Then a numerical IPP algorithm (we use the 4th order Runge-Kutta method [20]) is employed to predict the trajectory down to its

impact point. At the same time, the corresponding covariance is also predicted to the impact point using a zero-gain (open loop) EKF covariance equation. Then the 99% chi-square probability region ellipse is [2]

$$[\, x_{\text{ip}} - \hat{x}_{\text{pd}} y_{\text{ip}} - \hat{y}_{\text{pd}} \,] P_{\text{pd}}^{-1} \begin{bmatrix} x_{\text{ip}} - \hat{x}_{\text{pd}} \\ \\ x_{\text{ip}} - \hat{y}_{\text{pd}} \end{bmatrix} = \chi_2^2 (99\%) \tag{1.5.1}$$

where $\begin{pmatrix} x_{\text{ip}} & y_{\text{ip}} \end{pmatrix}$ is the true impact point, $\begin{pmatrix} \hat{x}_{\text{pd}} & \hat{y}_{\text{pd}} \end{pmatrix}$ is the predicted impact point, P_{pd} is the corresponding predicted covariance matrix, and $\chi_2^2 (99\%)$ denotes the 99% point on the chi-square cumulative distribution function with two degrees of freedom [2].

We use as "warning zone" the region (1.5.1) circumscribed by a circle (called the *uncertainty circle*)

$$[\, x_{\text{ip}} - \hat{x}_{\text{pd}} y_{\text{ip}} - \hat{y}_{\text{pd}} \,] \begin{bmatrix} \lambda_{\max} & 0 \\ \\ 0 & \lambda_{\max} \end{bmatrix}^{-1} \begin{bmatrix} x_{\text{ip}} - \hat{x}_{\text{pd}} \\ \\ x_{\text{ip}} - \hat{y}_{\text{pd}} \end{bmatrix} = \chi_2^2 (99\%) \tag{1.5.2}$$

where $\lambda_{\max}$ is the maximum eigenvalue of the covariance matrix P_{pd}. The uncertainty circle (1.5.2) can be used as the probability region for the true impact point, centered at the predicted impact point.

1.6 Real Data Results

The algorithm was evaluated on 30 cases of field-collected data with the same sampling frequency 30 Hz but different data lengths.

1.6.1 MIMM Approach

The MIMM estimator is chosen to consist of $L = 4$ IMM estimators with initial drag coefficient estimates $\hat{\alpha}(0)$: 0.18, 0.13, 0.065 and 0.035 (m^2/kg), respectively. The initial SD of each drag coefficient estimate is 25% of the corresponding initial estimate of the drag coefficient.

1.6.2 N-point Initialization and Measurement Noise Intensities

The initialization is crucial to the accuracy of the state estimates. One-point or two-point initializations are practical approaches for many cases [2]; however, with a high sampling rate, as in the present case, an N-point initialization is feasible, reasonable, and of far preferable accuracy. The method is based on the polynomial fitting of a set of noisy scalar measurements [2].

The N-point fitting of a polynomial of order n is done according to the model

$$\mathbf{z}(k) = H_{\mathbf{y}}(k)\mathbf{y} + \mathbf{w}(k) \stackrel{\Delta}{=} \begin{bmatrix} \mathbf{h}(k)' & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{h}(k)' & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{h}(k)' \end{bmatrix} \begin{bmatrix} \mathbf{y}_x \\ \mathbf{y}_y \\ \mathbf{y}_z \end{bmatrix} + \mathbf{w}(k)$$

where $k = 1, 2, \ldots, N$ and

$$\mathbf{h}(k) = \left[1 \;\; t_k \;\; \cdots \;\; \frac{(t_k)^n}{n!} \right]' \tag{1.6.1}$$

t_k is the sampling time and

$$\mathbf{y}_i = [\, a_{i0} \;\; a_{i1} \;\; \cdots \;\; a_{in} \,]' \qquad i = x,\; y,\; z \tag{1.6.2}$$

Assuming the projectile is moving with a nearly constant acceleration (NCA,

that is, $n = 2$) for the short time covered by N points, we get the estimate for the parameter vector $\mathbf{y}$ (of dimension $3(n+1) = 9$) as

$$\hat{\mathbf{y}} = \left[H_{\mathbf{y}}^{N\prime} \left(R^N \right)^{-1} H_{\mathbf{y}}^N \right]^{-1} H_{\mathbf{y}}^{N\prime} \left(R^N \right)^{-1} \mathbf{z}^N \tag{1.6.3}$$

with the corresponding covariance matrix

$$P_{\mathbf{y}} = \left[H_{\mathbf{y}}^{N\prime} \left(R^N \right)^{-1} H_{\mathbf{y}}^N \right]^{-1} \tag{1.6.4}$$

where

$$\mathbf{z}^N = \begin{bmatrix} \mathbf{z}(1) \\ \vdots \\ \mathbf{z}(N) \end{bmatrix} \tag{1.6.5}$$

is the stacked vector of measurements (of dimension $3N \times 1$),

$$H_{\mathbf{y}}^N = \begin{bmatrix} H_{\mathbf{y}}(1) \\ \vdots \\ H_{\mathbf{y}}(N) \end{bmatrix} \tag{1.6.6}$$

is the stacked measurement matrix (of dimension $3N \times 9$), and

$$R^N = \begin{bmatrix} R(1) \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots R(N) \end{bmatrix} \tag{1.6.7}$$

is the block diagonal covariance matrix of measurement noise (of dimension $3N \times 3N$) for the fitting interval.

Mapping the estimates (1.6.3) and the covariance (1.6.4) to the time of the most recent measurement [2], we get the initial estimates for position, velocity and acceleration. Based on the initial drag coefficient estimate and the acceleration estimates (last component of each direction, i.e., a_{i2}), the thrust will be initialized by calculating the acceleration in each direction (with the vertical one corrected by g).

Further, note that the goodness-of-fit error has the following chi-square distribution [2]

$$J_N^* \triangleq \left[\mathbf{z}^N - H_p^N \hat{\mathbf{y}}\right]' \left(R^N\right)^{-1} \left[\mathbf{z}^N - H_p^N \hat{\mathbf{y}}\right] \sim \chi^2_{Nn_{\mathbf{z}}-n_{\mathbf{y}}} \tag{1.6.8}$$

where $n_{\mathbf{z}} = 3$ is the dimension of $\mathbf{z}(i)$ and $n_{\mathbf{y}}$ is the dimension of $\mathbf{y}$ in our NCA model.

The N-point initialization approach with $N = 25$ is used (about 1s data in our case) and the measurement noise SD are assumed time-invariant with value 2 m for each coordinate (this choice was based on evaluating (1.6.8). Based on the initialization with the NCA model, the initial thrust estimate $\hat{\tau}(0)$ can be calculated and the corresponding initial estimate's SD can also be obtained. However, note that the NCA model is for thrust initialization only; each of these IMM estimators, with different initial drag coefficient estimates, starts from the beginning of measurements with two-point initialization [2] for position and

velocity.

1.6.3 Selection of Process Noise Intensities

In the dynamic equations, we actually assume a nearly constant velocity model for the kinematic component and a Wiener process with slow variation for the drag coefficient and thrust. In order to satisfy this model's assumptions, we need to choose small process noise "intensities" (PSD) q_v, q_α, and q_τ in following sense.

The process noise induced root mean square (RMS) rate of change in velocity over a sampling interval Δ is

$$d_v \triangleq \frac{\sqrt{q_v \Delta}}{\Delta} \ [(\text{m/s})/\text{s}] \tag{1.6.1}$$

For the drag coefficient the rate of change is

$$d_\alpha \triangleq \frac{\sqrt{q_\alpha \Delta}}{\Delta} \ [(\text{m}^2/\text{kg})/\text{s}] \tag{1.6.2}$$

For the thrust the rate of change is

$$d_\tau \triangleq \frac{\sqrt{q_\tau \Delta}}{\Delta} \ [(\text{m/s}^2)/\text{s}] \tag{1.6.3}$$

Then we have the following estimator design procedure for selection of the

process noise PSD based on the corresponding RMS rates of change

$$q_v = d_v^2 \Delta \quad (\mathrm{m}^2/\mathrm{s}^3) \tag{1.6.4}$$

$$q_\alpha = d_\alpha^2 \Delta \quad [\mathrm{m}^4/(\mathrm{kg}^2 \cdot \mathrm{s})] \tag{1.6.5}$$

$$q_\tau = d_\tau^2 \Delta \quad (\mathrm{m}^2/\mathrm{s}^5) \tag{1.6.6}$$

The process noise intensities are chosen based on process noise induced RMS change (in velocity/ drag coefficient/ thrust) over an interval of 1 s shown in Table 1.6.2.

Table 1.6.2: RMS change rate due to process noise (TM and BM)

Filter	d_v [(m/s)/s]	d_α [(m^2/kg)/s]	d_τ [(m/s^2)/s]
IMM (TM)	7.5	$0.1\hat{\alpha}(0)$/s	$0.25\,\hat{\tau}(0)$/s
IMM (BM)	4.3	$0.08\hat{\alpha}(0)$/s	N/A

1.6.4 IMM Mode Transition Probability Matrix

The transition probability matrix for the two-mode IMM is

$$\pi = \begin{bmatrix} p_{11} & 1 - p_{11} \\ 1 - p_{22} & p_{22} \end{bmatrix} \tag{1.6.1}$$

where the elements of the matrix are obtained based on the mean sojourn time (MST) [2], s_1 and s_2, for the TM ($i = 1$) and the BM ($i = 2$) respectively

$$p_{ii} = 1 - \frac{T}{s_i} \qquad i = 1, 2 \tag{1.6.2}$$

and $T = 1/30$ s is the (fixed) sampling interval. Setting the MST as $s_1 = 2$ s and

$s_2 = 50\,\mathrm{s}$ for TM and BM respectively, we get the mode transition probability matrix

$$\pi = \begin{bmatrix} 0.9833 & 0.0167 \\ 0.0007 & 0.9993 \end{bmatrix} \tag{1.6.3}$$

with initial mode probability vector $[\,0.90 \;\; 0.10\,]$.

1.6.5 Real Data Results

A sample estimated trajectory, indicating the portions corresponding to the MIMM filtering and to the IPP (including the uncertainty circle centered at the IPP), is presented in Fig. 1.6.1. The estimated burnout point (BoP) is also indicated.

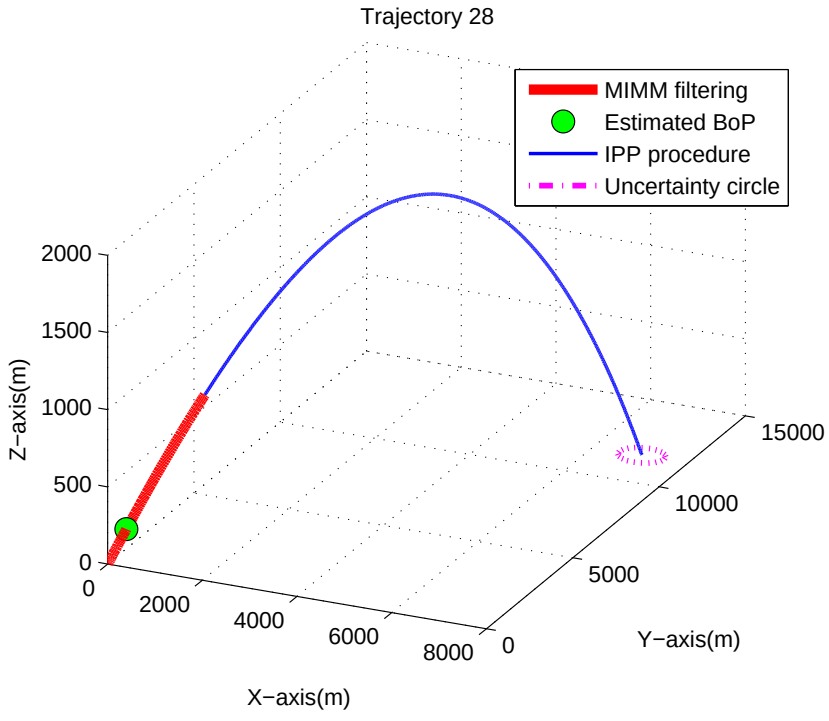


Figure 1.6.1: A sample estimated trajectory (traj. 28).

The IPP performance of 30 real trajectories for short range, medium range and long range are presented Tables 1.6.3–1.6.5, respectively, and the following acronyms are used:

- #: trajectory ID, with indication of the projectile ranges S, M, L (Short, Medium, Long)
- Rprd: IPP range (m)
- Ract: actual range (m) — *this was the only impact point information (ground truth) when the data were recorded*
- Rerr: IPP range error (m)
- Rerr/R: ratio between IPP range error and actual range (*IPP error percentage*)
- Rad/R: ratio between the uncertainty radius and actual range (*uncertainty circle percentage*)
- A/B/C: {IPP error percentage less than 10%}/{uncertainty circle percentage less than 10%}/{both A and B}
- M: Major axis of uncertainty ellipse, also the radius of uncertainty circle (m)
- m: minor axis of uncertainty ellipse (m)
- DrgIni: initial drag coefficient estimate (m^2/kg) of the best IMM
- DrgEnd: drag coefficient estimate at the end of the observation interval (m^2/kg)
- ThrIni: initial thrust estimate (m/s^2) of the best IMM
- Vmax: maximum speed (velocity magnitude) estimate (m/s) and the sampling point where it happened

BoP: burnout point – based on the best IMM's TM probability

TmP/TtP total measurement points / total trajectory (measurement+prediction) points (and percentage).

The *burnout point* is decided based on the best IMM's mode probabilities. The point, from which, the mode probability of TM stays lower than 0.03 for 10 consecutive sample points (i.e., BM dominates), is recognized as the burnout point. After the recognized burnout point plus the 10 sample points for deciding the burnout point, we stop using TM and start using an additional ballistic mode, say BM1, with a larger process noise setting to cover possibly higher uncertainties (projectiles may experience tumbling or high measurement uncertainties due to change of the relative sensor-target geometry). The BM1 has the process noise induced RMS change rate in velocity as $2d_v$ and the same other settings as those of the BM. The mode transition probability matrix between BM and BM1 is

$$\pi = \begin{bmatrix} 0.70.3 \\ \\ 0.30.7 \end{bmatrix} \tag{1.6.1}$$

with initial mode probability vector [0.90 0.10].

Table 1.6.3: IPP performance for 5 real trajectories (short range)

#	Rprd	Ract	Rerr	Rerr/R	Rad/R	A / B / C	M	m	DrgIni	DrgEnd	ThrIni	Vmax	BoP	TmP/TtP
(S)06	7203	7541	-338	4.48	3.67	T / T / F	277	146	0.13	0.093	378	492(38)	41	150/883(17.0%)
(S)22	5092	5937	-845	14.23	4.6	F / T / F	273	227	0.13	0.194	274	394(39)	45	150/1235(12.1%)
(S)25	6650	6642	8	0.12	6.91	T / T / T	460	290	0.18	0.113	302	408(35)	40	139/1358(10.2%)
(S)29	6333	6747	-414	6.14	6.27	T / T / T	424	286	0.18	0.119	335	412(33)	39	150/1355(11.1%)
(S)30	4811	4884	-73	1.49	9.9	T / T / T	484	374	0.13	0.125	293	406(36)	40	117/1567(7.5%)

Table 1.6.4: IPP performance for 15 real trajectories (medium range)

#	Rprd	Ract	Rerr	Rerr/R	Rad/R	A / B / C	M	m	DrgIni	DrgEnd	ThrIni	Vmax	BoP	TmP/TtP
(M)01	6544	8932	-2388	26.74	4.43	F / T / F	396	245	0.13	0.118	325	458(41)	44	150/1233(12.2%)
(M)05	8729	9608	-879	9.15	4.46	T / T / F	429	286	0.065	0.088	479	552(27)	28	150/1347(11.1%)
(M)07	7934	9041	-1107	12.24	7.24	F / T / F	655	274	0.13	0.066	135	361(59)	62	149/1252(11.9%)
(M)08	8540	9041	-501	5.54	7.32	T / T / T	663	267	0.13	0.092	283	497(34)	37	101/1265(8.0%)
(M)09	8872	9020	-148	1.64	4.75	T / T / T	429	239	0.065	0.086	460	550(40)	43	138/1200(11.5%)
(M)10	8669	10067	-1398	13.89	5.12	F / T / F	516	242	0.065	0.091	342	515(39)	42	113/1203(9.4%)
(M)11	7829	9173	-1344	14.65	4.31	F / T / F	396	228	0.18	0.119	331	504(43)	47	138/1206(11.4%)
(M)12	8873	9023	-150	1.66	4.67	T / T / T	421	277	0.065	0.089	394	485(32)	35	150/1329(11.3%)
(M)13	7869	8133	-264	3.25	5.89	T / T / T	479	292	0.13	0.093	322	408(37)	43	149/1358(11.0%)
(M)14	9338	9338	0	0	5.09	T / T / T	476	305	0.065	0.08	413	522(33)	35	150/1399(10.7%)
(M)15	10336	8696	1640	18.86	8.36	F / T / F	728	337	0.13	0.065	364	503(38)	40	126/1453(8.7%)
(M)19	10412	9671	741	7.66	9.54	T / T / T	923	368	0.13	0.065	317	492(39)	43	110/1523(7.2%)
(M)20	8666	8258	408	4.94	5.77	T / T / T	477	316	0.065	0.1	394	521(34)	37	127/1448(8.8%)
(M)21	8150	8177	-27	0.33	5.47	T / T / T	448	257	0.13	0.083	310	438(41)	46	178/1261(14.1%)
(M)27	10070	9873	197	2	6.68	T / T / T	660	320	0.18	0.068	410	540(39)	41	150/1423(10.5%)

Table 1.6.5: IPP performance for 10 real trajectories (long range)

#	Rprd	Ract	Rerr	Rerr/R	Rad/R	A / B / C	M	m	DrgIni	DrgEnd	ThrIni	Vmax	BoP	TmP/TtP
(L)02	22765	20813	1952	9.38	8.89	T / T / F	1851	718	0.035	0.033	276	664(68)	73	139/2385(5.8%)
(L)03	14708	14585	123	0.84	4.43	T / T / T	646	299	0.065	0.045	390	670(48)	50	152/1398(10.9%)
(L)04	16849	13409	3440	25.65	10.05	F / F / F	1348	370	0.18	0.033	388	619(63)	69	144/1566(9.2%)
(L)16	16220	15740	480	3.05	5.16	T / T / T	813	326	0.035	0.036	342	669(51)	54	140/1458(9.6%)
(L)17	17136	16405	731	4.46	7.26	T / T / T	1191	461	0.035	0.044	350	616(50)	52	114/1805(6.3%)
(L)18	16474	17742	-1268	7.15	7.18	T / T / T	1274	455	0.065	0.05	393	688(46)	50	109/1804(6.0%)
(L)23	17494	17076	418	2.45	6.21	T / T / T	1061	453	0.035	0.039	331	671(31)	34	113/1768(6.4%)
(L)24	16031	16196	-165	1.02	6.32	T / T / T	1025	385	0.035	0.047	325	656(45)	50	113/1620(7.0%)
(L)26	16050	16871	-821	4.87	8.57	T / T / T	1447	416	0.13	0.046	329	616(63)	81	138/1705(8.1%)
(L)28	13624	13743	-119	0.87	3.46	T / T / T	476	235	0.035	0.041	345	631(36)	38	155/1198(12.9%)

The evolution of the drag coefficient estimates in the MIMM is shown in Figs. 1.6.2 and 1.6.3 (for trajectories 12 and 28, respectively). It can bee seen that in the MIMM estimator the drag coefficient estimates of different IMM estimators converge toward to the end of the observation period. Based on (1.4.6), the best IMM estimator (with the highest likelihood) is selected.

The drag coefficient and thrust estimates in the best IMM estimator selected using (1.4.6) are shown in Figs. 1.6.4 and 1.6.5 (for trajectories 12 and 28, re-

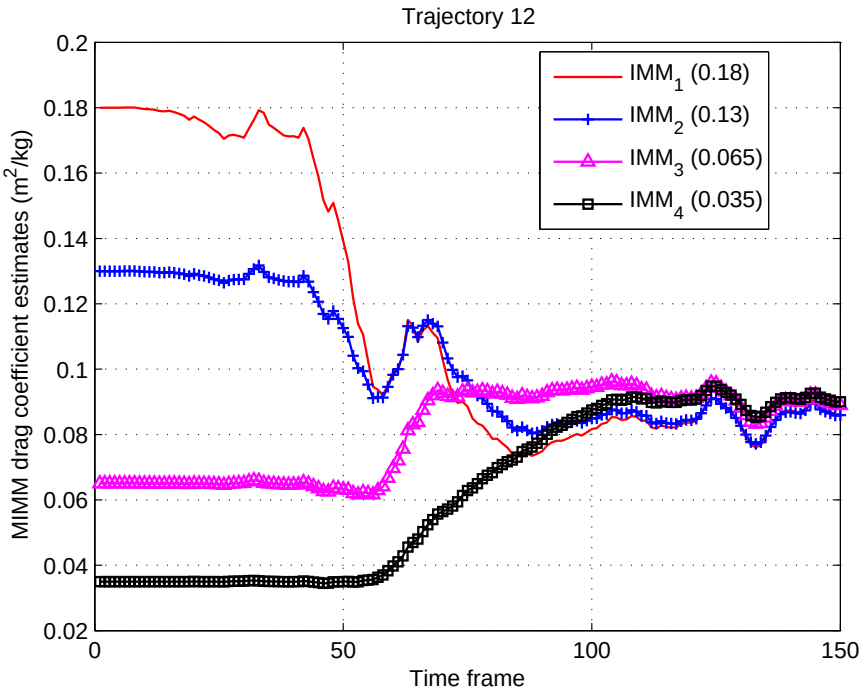


Figure 1.6.2: Evolution of the drag coefficient estimates in the MIMM (trajectory 12)

spectively). The ambiguity between the drag coefficient and thrust can be observed. A sudden decrease in the drag coefficient estimate (as in Fig. 1.6.5 at time $k = 17$) triggers an increase in the thrust estimate and vice versa.

The estimation ambiguity between the drag coefficient and thrust makes the unbiased mixing strategy a key to correctly estimating the thrust component in the IMM estimator. To illustrate this ambiguity in more detail, consider a comparison between the IMM with the conventional (biased) mixing and with the unbiased mixing. Figs. 1.6.6 and 1.6.7 show the drag coefficient and thrust estimates obtained with the conventional mixing and the unbiased mixing for trajectory 21, respectively. Fig. 1.6.6 shows that, because of the biased estimates

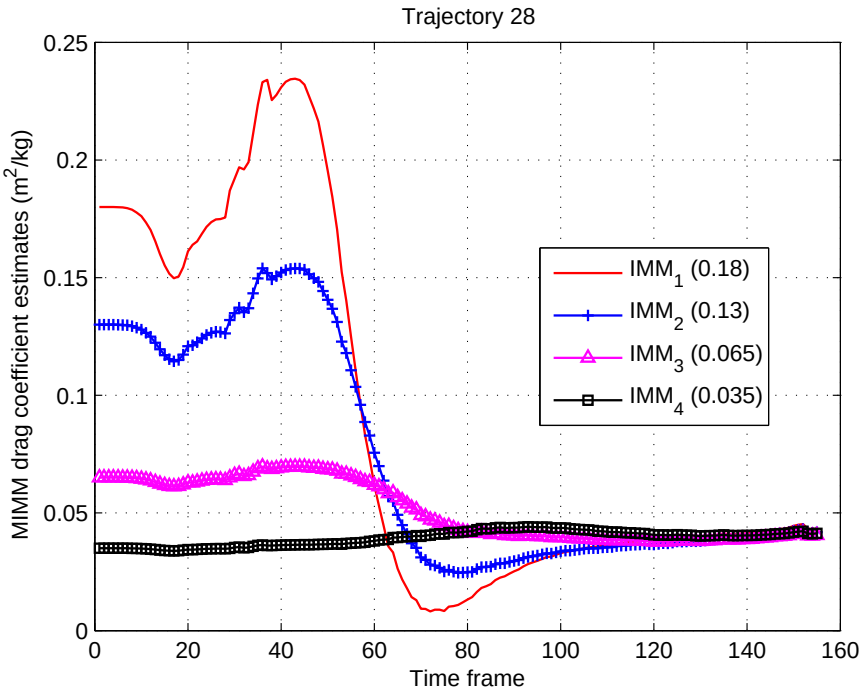


Figure 1.6.3: Evolution of the drag coefficient estimates in the MIMM (trajectory 28)

of the thrust, the drag coefficient estimates try to go negative to compensate for the thrust estimate that became practically zero (the algorithm implementation forced them to stay nonnegative). Note that in the IMM with conventional mixing, the TM is discontinued prematurely at $k = 30$ (about 1 s after target acquisition) because the thrust estimate is zero. This is "compensated" by a negative drag coefficient estimate, which was, however, forced to be nonnegative. In Fig. 1.6.7, it can be seen that TM is discontinued at $k = 56$ (about 2 s after target acquisition). This was based on the TM probability being less than 0.03 for 10 consecutive samples. The drag coefficient estimates stabilize around 0.09 m^2/kg. Thus, the unbiased mixing can be seen to avoid the tendency of

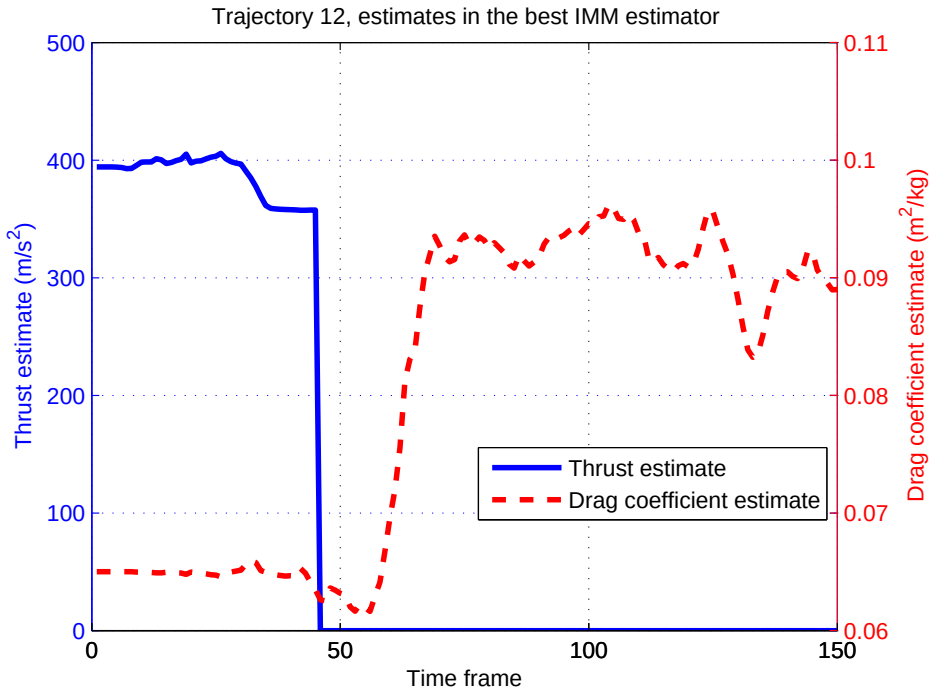


Figure 1.6.4: Drag coefficient and thrust estimates in the best IMM estimator (trajectory 12). Note that TM is discontinued at $k = 45$.

the drag coefficient estimate to become negative to compensate for the thrust estimate that becomes biased towards zero.

The IPP performance is evaluated by the three criteria (A/B/C) as shown in Table 1.6.6. A single IMM with conventional mixing using one initial drag coefficient estimate 0.1 m^2/kg with a large initial SD of 0.08 m^2/kg is employed as a performance comparison baseline. It can be seen that compared with a single IMM, the MIMM-UM approach gives a significantly improved estimation performance, with smaller IPP errors and smaller uncertainties.

For the real data, if LoS measurement accuracy and the sensor-target geometry would have been available, the use of the technique of [18] to obtain $R(k)$ as a

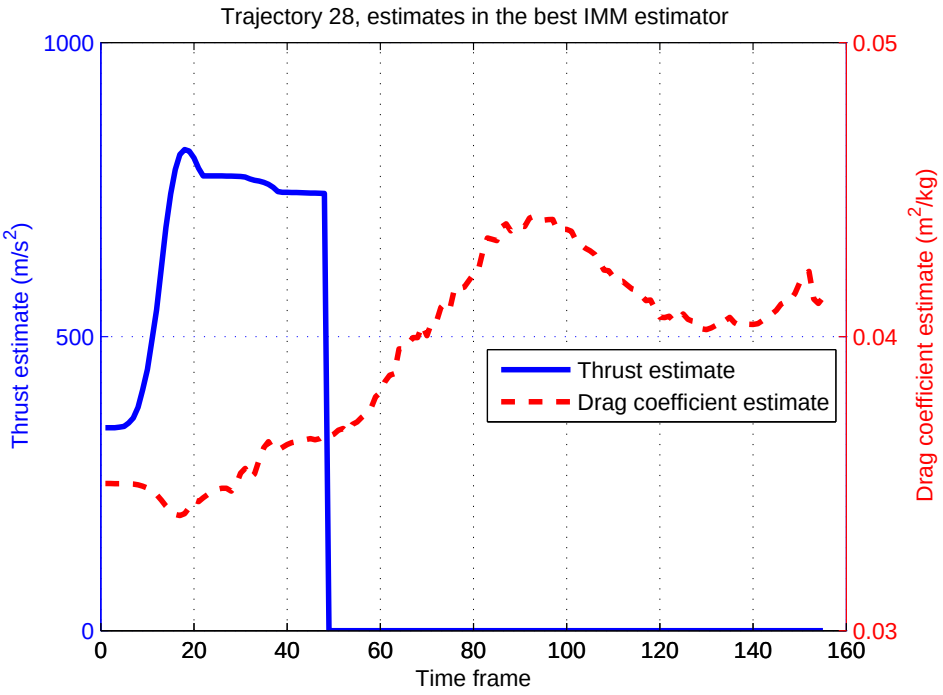


Figure 1.6.5: Drag coefficient and thrust estimates in the best IMM estimator (trajectory 28). Note that TM is discontinued at $k = 48$.

time varying covariance matrix would have allowed overall better tracking and IPP performance. Another difficulty in working with the available real data is the short span of the observation (5–15% of the trajectory) and the possibility of ill-conditioning of the LoS observations.

Table 1.6.6: IPP performance evaluation summary

Criterion	% of cases	
	MIMM-UM	Conventional Single IMM
A	23/30 (77%)	19/30 (63%)
B	29/30 (97%)	08/30 (27%)
C	20/30 (67%)	23/30 (77%)
B & C	**20/30** (67%)	04/30 (13%)

1.7 Conclusions

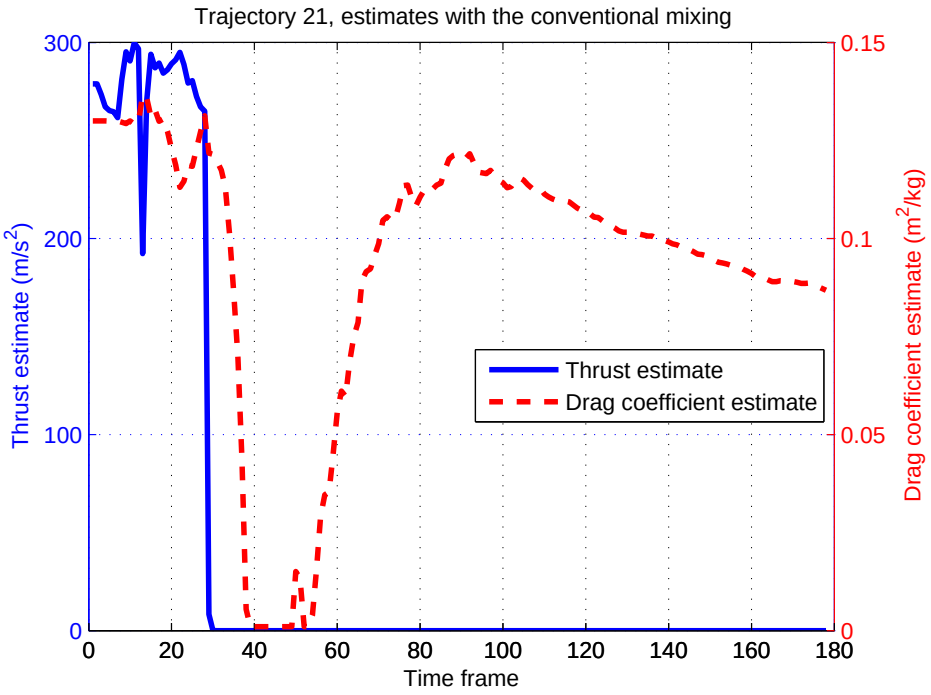


Figure 1.6.6: Drag coefficient and thrust estimates (in the best IMM estimator, trajectory 21) with the conventional mixing.

Facing the challenges of very short time span of observations (about 10% of the whole trajectory) on thrusting projectiles, the MIMM estimator with different initial drag coefficient estimates is presented to alleviate the ambiguity in the estimation of the drag coefficient and thrust and to obtain more accurate estimates of the drag coefficient and thrust. A novel unbiased mixing approach for an IMM estimator with modes that have unequal dimension state vector is introduced and is an important extension of the conventional mixing in an IMM estimator. This gives an unbiased estimate of the thrust component and leads to significant improvement for the final IPP performance. The best IPP performance was achieved by employing the MIMM estimator with the novel

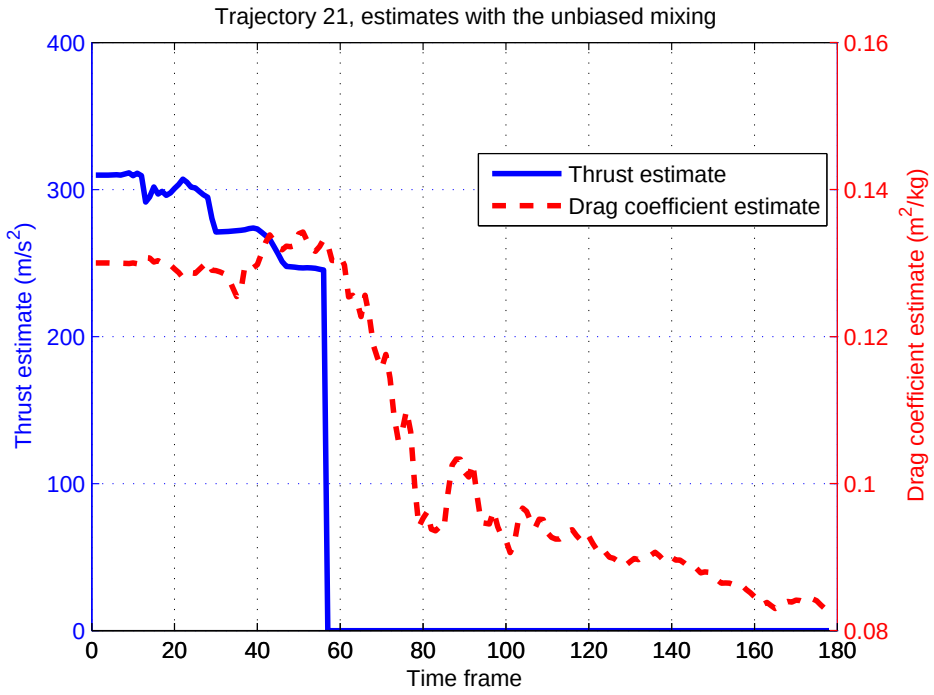


Figure 1.6.7: Drag coefficient and thrust estimates (in the best IMM estimator, trajectory 21) with the unbiased mixing.

unbiased mixing.

Chapter 2

Impact Point Prediction for Thrusting Projectiles in the Presence of Wind

Wind can and often does significantly affect impact-point prediction (IPP) performance for thrusting/ballistic endoatmospheric projectiles. Wind exacerbates the estimation ambiguity between drag and thrust in the dynamic model and induces additional uncertainty in the IPP procedure. A tracker accounting for the wind effect is presented and simulation study shows that it can be fully compensated if the wind information is available. An N-point adaptive initialization based on a goodness-of-fit test and a statistical significance test is introduced. Based on the multiple interacting multiple model (MIMM) approach developed in the former Chapter, the IPP performance is investigated with respect to the total observation time and the sensor accuracy in various wind scenarios.

In each Monte Carlo (MC) run of the simulation study, under the same sensor accuracy and the same observation time, the same set of random numbers has been used (but different in different MC runs) for the same caliber projectile in various wind scenarios to examine how much the wind affects the IPP performance with/without the exact knowledge of the wind information. The

final conclusion is that with the wind effect accounted for, the IPP performance in the presence of wind is practically the same as in its absence.

2.1 Introduction

Impact point prediction (IPP) of a thrusting/ballistic projectile is significantly affected by wind. The wind effect, in addition to the short available observation time and the limited sensor accuracy, is an issue of high importance and concern.

The wind effect causes the moving projectile to turn into the "apparent" wind, i.e., it causes the nose of the moving projectile to be pointed into the wind as observed from the projectile's (moving) frame of reference [14] [37]. A tail- or head- wind, if unaccounted for, would cause the IPP to give under- or over-prediction for the projectile's range (hence the term range wind effect); a cross wind would lead to left- or right-deviation. The vertical wind is ignored in this study, as it is typically not present. Consequently, the projectile's dynamic model is modified from [31] [27] to incorporate the wind effect. Based on the modified model, the sensitivity of the IPP performance with respect to the wind effect is studied.

The available observation time is a key factor that significantly affects the IPP performance and becomes even more critical in the presence of wind [14] [22]. For the IPP problem, using a state model with drag coefficient and thrust estimated as separate components, there is an important observation-time tradeoff

between accuracy and implementation of countermeasures. Before the prediction procedure starts, the collected measurements are used to initialize the estimator, attack the estimation ambiguity between the drag coefficient and the thrust components, and the probability of the correct mode should converge close to unity [27]. The wind effect could impact each of these and evidently deteriorate the overall IPP performance.

Due to the nonlinear nature of the dynamics of the thrusting/ballistic projectiles, linearization and discretization of the nonlinear system is an approximation of reality. For a fixed sampling rate, the approximation quality is determined by the sensor accuracy. The observation uncertainty caused by the sensor errors (in range and azimuth/elevation angles) affects the estimation results, as well as the prediction procedure that follows, and hence the final IPP performance. Wind uncertainty exacerbates the observation uncertainty.

The present chapter aims to offer an extensive analysis of the wind effect on the recently developed IPP system using a multiple interacting multiple model (MIMM) estimator (with different initial drag coefficient estimates and using unbiased mixing[3]) [27]. As an important practical issue, in many situations the wind information is not available. To examine how much effect this has on the IPP performance, three situations are considered, i.e., given perfect knowledge of the wind information (denoted concisely as "g.w.i." for later

[3]The unbiased mixing of IMM estimator is the key for correct estimation of the extra components when the mode-matched filters of IMM estimator are of unequal dimensions, e.g., the thrust is the extra component in the present discussion.

use), given no knowledge of the wind information ("n.w.i."), given the wind information with certain deterministic error ("e.w.i."). In order to provide a comprehensive insight, the IPP performance is investigated with respect to the total observation time and the sensor accuracy under various wind scenarios. It is worth noting that, for simplicity, we assume a constant wind during the estimation and prediction procedure (i.e., the whole trajectory period). Altitude/location-dependent wind can be easily dealt with in the same manner. An N-point adaptive initialization based on a goodness-of-fit test and a statistical significance test is introduced. For each IMM estimator (associated with a selected drag coefficient) from the MIMM estimator, the initialization of the thrust component is of special importance when only a very short total observation time is available. A good initialization will alleviate the estimation ambiguity between the drag coefficient and the thrust. The N-point adaptive initialization particularly provides an early and reasonably accurate estimate between the drag coefficient and thrust (as well as the kinematic components) and leads to a quick identification of the correct mode of the IMM estimator.

The chapter is organized in following manner. In Section 2.2, the wind effect is presented and under a flat-Earth assumption (suitable for short range projectiles), the dynamic model in the *presence of wind* is modified from [27]. The corresponding discretized form of the modified dynamic model and the discrete-time measurement equations are also presented here. The MIMM estimator and the IPP procedure are described in Section 2.3. The parameter

setting, design of the MIMM estimator and the N-point adaptive initialization approach are presented in Section 2.4. Then in the presence of different wind (strength and direction) conditions, the IPP performance is investigated in various total observation time and sensor accuracy scenarios by simulation in Section 2.5. Conclusions are presented in Section 2.6.

2.2 Dynamic Model in the Presence of Wind

The wind effect, which worsens the estimation ambiguity between the drag coefficient and thrust [27] and thus presents an additional challenge to the IPP, must be carefully accounted for [4]. The sensitivity of the IPP performance to the drag coefficient estimate, which is significantly affected by the relative velocity of the projectile (even one with known shape) with respect to air, necessitates the quantification of the wind effect for the purpose of IPP.

The contribution of the wind comprises the range wind, cross wind and vertical wind effects. We ignore the last, since it is generally small; but the techniques we present could be augmented to account for it. A range (head/tail) wind will push back or forward the moving object and a cross wind causes the moving object deviate to the side. These wind components together are considered as the true wind velocity and are combined with the target velocity with respect to the ground to yield the so-called "apparent wind". The moving target turns into the apparent wind, i.e., its nose is pointed into the wind (or "upwind") while its tail pointing "downwind" [14] [37]. With the wind shown in Fig. 2.2.1,

we have

$$\mathbf{v}_a = \mathbf{v}_w - \mathbf{v} \tag{2.2.1}$$

where $\mathbf{v}$ is the target velocity with respect to the ground, $\mathbf{v}_w$ is the wind velocity and $\mathbf{v}_a$ is the apparent wind velocity.

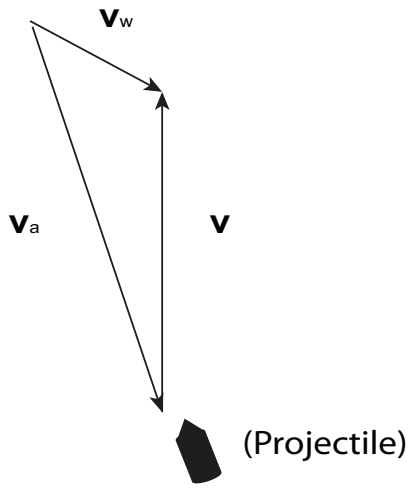


Figure 2.2.1: The wind effect ($\mathbf{v}$ is the target's ground velocity, $\mathbf{v}_w$ is the wind velocity and $\mathbf{v}_a$ is the apparent wind velocity).

Thus, the drag and the thrust in the following dynamic model should be aligned with the direction of the projectile, which is aligned with the apparent wind. In the absence of information about how fast the projectiles align themselves with the apparent wind following launch, it is assumed that this happens instantaneously. During thrusting phase this is especially important, as due to wind thrust may thus not be aligned with the direction of travel.

For simplicity of analysis, we assume a constant wind during the estimation and

prediction procedure (i.e., the whole trajectory period). The wind environment where the instantaneous projectile alignment hypothesis is reasonable and acceptable (e.g., altitude/location-dependent wind with slow variation or time-variant wind that is piecewise constant with respect to altitude) can be dealt with in the same manner.

In the presence/absence of wind, the trajectory of a thrusting projectile, from launch to impact, can be divided into two phases: thrusting and ballistic. It is a natural choice that we use an IMM estimator with a thrust mode (TM) and a ballistic mode (BM) to match these phases. The state vector for the corresponding TM is

$$\mathbf{x}(t) = [\, x(t)\ y(t)\ z(t)\ \dot{x}(t)\ \dot{y}(t)\ \dot{z}(t)\ \alpha(t)\ \tau(t)\,]' \tag{2.2.2}$$

where $\alpha(t)$ is the drag coefficient and $\tau(t)$ is the thrust. The state vector of the BM is the same as above but excludes the thrust component. The time arguments will be omitted where no ambiguity is caused.

Under flat-Earth assumption, the dynamic model *in the presence of wind* modified from [27] (the corresponding modifications are indicated by the sub-script "a", which implies the use of "apparent" wind information) can be written as follows

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{\tau}{V_a} \begin{bmatrix} \dot{x}_a \\ \dot{y}_a \\ \dot{z}_a \end{bmatrix} + \alpha\, \alpha_m\, D_a \begin{bmatrix} \dot{x}_a \\ \dot{y}_a \\ \dot{z}_a \end{bmatrix} + \mathbf{g} + \tilde{\nu}_1 \tag{2.2.3}$$

and

$$\dot{\alpha} = \tilde{\nu}_2 \tag{2.2.4}$$

$$\dot{\tau} = \tilde{\nu}_3 \tag{2.2.5}$$

where

- The target velocities with respect to air are $\dot{x}_a \stackrel{\Delta}{=} \dot{x} - \dot{x}_w$, $\dot{y}_a \stackrel{\Delta}{=} \dot{y} - \dot{y}_w$ and $\dot{z}_a \stackrel{\Delta}{=} \dot{z} - \dot{z}_w$.
- The first term on the right side of (2.2.3) is the specific thrust in the x, y, and z directions with accounting for the wind effect. For the ballistic phase, the thrust is zero.
- The second term is the drag part, which is related to the altitude and target velocity with respect to air.
- V_a is the magnitude of the velocity vector $[\, \dot{x}_a \quad \dot{y}_a \quad \dot{z}_a \,]'$, i.e., the speed (in m/ s).
- α is the drag coefficient (in m^2/kg) at subsonic speed and τ is the thrust (in m/s^2). It is known that the drag coefficient varies significantly with the Mach number regime: subsonic, transonic and supersonic. This will be accounted for by the α_m as below.
- α_m is the (dimensionless) *Mach number-dependent drag coefficient multiplier*, which is approximated by a cubic spline curve shown in Fig. 1.3.1 (of Chapter 1).

- $D_a = -\frac{\rho(z)V_a}{2}$, where $\rho(z) = \rho_0 e^{-cz}$ is the air density (in kg/m^3) at altitude z (in m) and c is the air density constant (in m^{-1}) [21].
- $\mathbf{g} \stackrel{\Delta}{=} [\,0 \;\; 0 \;\; -g\,]'$ is the standard gravity vector. g is the standard acceleration due to gravity at sea level, assumed to be the same throughout the trajectory, with value 9.812 m/s^2.
- $\tilde{v}_1$, $\tilde{v}_2$, and $\tilde{v}_3$ are assumed to be independent continuous-time zero-mean white Gaussian noises. The drag coefficient and thrust acceleration are thus modeled as Wiener processes with slow variation [2].

Combining the dynamic equations (2.2.3)–(2.2.5), we have the following compact form (accounting for the wind effect)

$$\dot{\mathbf{x}}(t) = f_w[\mathbf{x}(t),\ \mathbf{x}_w(t)] + \tilde{v}(t) \tag{2.2.6}$$

where

$$f_w[\mathbf{x}(t),\ \mathbf{x}_w(t)] = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \\ \tau(t)\dfrac{\dot{x}_a(t)}{V_a(t)} + \alpha(t)\alpha_m(t)D_a(t)\dot{x}_a(t) \\ \tau(t)\dfrac{\dot{y}_a(t)}{V_a(t)} + \alpha(t)\alpha_m(t)D_a(t)\dot{y}_a(t) \\ \tau(t)\dfrac{\dot{z}_a(t)}{V_a(t)} + \alpha(t)\alpha_m(t)D_a(t)\dot{z}_a(t) - g \\ 0 \\ 0 \end{bmatrix} \tag{2.2.7}$$

with the wind effect compensation vector

$$\mathbf{x}_w(t) = [\, 000\dot{x}_w(t)\dot{y}_w(t)\dot{z}_w(t)00 \,]' \tag{2.2.8}$$

and the Wiener process vector

$$\tilde{v}(t) = [\, \tilde{v}_1(t)' \;\; \tilde{v}_2(t) \;\; \tilde{v}_3(t) \,]' \tag{2.2.9}$$

Note that the last two components of (2.2.7) are both equal to zero, as we assume the drag coefficient and thrust are Wiener processes.

The state vector equation is discretized by a second order Taylor expansion[4] [15]. Using the discrete time notation $\mathbf{x}(k) = \mathbf{x}(t)|_{t=kT}$ and $\mathbf{x}(k+1) = \mathbf{x}(t+T)|_{t=kT}$ (the same for $\mathbf{x}_w(t)$), we have the following discretized continuous-time expression [27]

$$\mathbf{x}(k+1) = \mathbf{x}(k) + f_w[\mathbf{x}(k), \mathbf{x}_w(k)]T + A_w(k) f_w[\mathbf{x}(k), \mathbf{x}_w(k)]\frac{T^2}{2} + v(k) \tag{2.2.10}$$

where $A_w(k)$ is the Jacobian of (2.2.7) evaluated at $\mathbf{x}(k)$ (with known wind information $\mathbf{x}_w(k)$) and $v(k)$ is the discretized continuous-time (white Gaussian) process noise for the sampling interval T. Based on the assumption that α is nearly constant and D_a is related to both z and V_a, the detailed form of A_w here is obtained by replacing in the Jacobian in Appendix C the target velocities

[4] It has been reported that for a short range ballistic projectile the first order Taylor expansion is sufficient. However, for the thrusting projectile with unknown drag coefficient and unknown thrust – as in the present study – the second order Taylor expansion is used to compensate for the nonlinearity made obscure by the unknowns. The methodology can be used for long range scenario, in which case a second order Taylor expansion is necessary to avoid bias [3].

(with respect to ground) with the corresponding target velocities with respect to air.

The corresponding covariance matrix of the discretized process noise is

$$Q = \begin{bmatrix} \begin{bmatrix} \frac{T^3}{3} I_3 & \frac{T^2}{2} I_3 \\ \frac{T^2}{2} I_3 & T I_3 \end{bmatrix} q_v & \mathbf{0}_{6\times 1} & \mathbf{0}_{6\times 1} \\ \mathbf{0}_{1\times 6} & T q_\alpha & 0 \\ \mathbf{0}_{1\times 6} & 0 & T q_\tau \end{bmatrix} \tag{2.2.11}$$

where I_3 is the 3×3 identity matrix and the continuous time process noise "intensities" q_v, q_α and q_τ are the corresponding power spectral densities (PSD).

Assuming the sensor is located at $[\, x_s \;\; y_s \;\; z_s \,]$, the measurements in spherical coordinates are

$$r_m = r + w_r = \sqrt{(x - x_s)^2 + (y - y_s)^2 + (z - z_s)^2} + w_r \tag{2.2.12}$$

$$\theta_m = \theta + w_\theta = \tan^{-1}\left(\frac{y - y_s}{x - x_s}\right) + w_\theta \tag{2.2.13}$$

$$\epsilon_m = \epsilon + w_\epsilon = \tan^{-1}\left(\frac{z - z_s}{\sqrt{(x - x_s)^2 + (y - y_s)^2}}\right) + w_\epsilon \tag{2.2.14}$$

where r, θ and ϵ are the independent true range, azimuth and elevation components, respectively. w_r, w_θ and w_ϵ denote the corresponding zero-mean white Gaussian measurement noises with standard deviations (SD) σ_r, σ_θ and σ_ϵ, respectively.

An unbiased measurement conversion from spherical to Cartesian coordinates was presented [3], so that the IPP problem can be described entirely in Cartesian

coordinates:

$$\mathbf{z}(k) \triangleq H\mathbf{x}(k) + \mathbf{w}(k) \tag{2.2.15}$$

where $H = [\, I_3 \quad \mathbf{0} \,]$, $\mathbf{w}(k)$ is the equivalent measurement noise vector in Cartesian coordinates obtained from the unbiased measurement conversion and $R(k)$ is the corresponding equivalent *state-dependent* covariance matrix.

2.3 Multiple-IMM Estimator for IPP

During the thrust mode (TM), the drag parameter (drag coefficient) and thrust parameter are separate state components as shown in (2.2.2). However, the drag force and thrust force are acting simultaneously (see equation (2.2.3)) and the IMM estimator has difficulty distinguishing between them if the initial uncertainty in the drag coefficient is large. A sudden decrease in the drag coefficient estimate may trigger an increase in the thrust estimate. In the presence of wind, the wind-induced motion makes correctly unsnarling these important components (drag coefficient and thrust) increasingly obdurate. The sensitivity of the estimation to the initial drag coefficient estimate necessitates the use of an MIMM estimator to overcome this "marginal observability" problem [2]. The procedure starts by establishing a set of the L IMM estimators, each with an appropriate set of modes (TM and BM in present case) to describe the system behavior. Each IMM estimator will be initialized with a different value of the drag coefficient estimate with a suitable initial SD. The filtering parameter

setting is discussed in Section 2.4.

In order to select the best initial drag coefficient estimate, we need to determine the most likely among L IMM estimators, during the observation period. The likelihood function of IMM estimator l for the time interval $[k_0, K]$ is

$$\Lambda_l^{k_0,K} = \prod_{k=k_0}^{K} \Lambda_l(k) \qquad l = 1, \ldots, L \tag{2.3.1}$$

where $\Lambda_l(k)$ is the likelihood function of lth IMM estimator, with r modes, at time k, given by [3]

$$\Lambda_l(k) \stackrel{\Delta}{=} \sum_{i=1}^{r} \Lambda_l^i(k) \mu_l^i(k|k-1) \tag{2.3.2}$$

where $\Lambda_l^i(k)$ is the likelihood function of mode i of IMM estimator l at time k and $\mu_l^i(k|k-1)$ is the predicted mode probability for mode i of IMM estimator l. The mode likelihood is [2]

$$\begin{aligned} \Lambda_l^i(k) &= p\left[\mathbf{z}(k)|M_l^i(k), \hat{\mathbf{x}}_l^{0i}(k-1|k-1), P_l^{0i}(k-1|k-1)\right] \\ &= \mathcal{N}\left[\mathbf{z}(k); \hat{\mathbf{z}}_l^i(k|k-1), S_l^i(k)\right] \end{aligned} \tag{2.3.3}$$

where $\mathcal{N}[\cdot]$ is the Gaussian probability density function, $\hat{\mathbf{z}}_l^i$ is the predicted measurement and S_l^i is the innovation covariance in mode i of IMM estimator

l. The predicted mode probability can be written as

$$\begin{aligned}\mu_l^i(k|k-1) &= \sum_{j=1}^{r} P\left\{M_l^i(k)|M_l^j(k-1), Z^{k-1}\right\} P\left\{M_l^j(k-1)|Z^{k-1}\right\} \\ &\triangleq \sum_{j=1}^{r} p_{ji}\mu_l^j(k-1) \end{aligned} \tag{2.3.4}$$

where $\mu_l^j(k-1)$ is the mode probability of mode j of IMM estimator l at time $k-1$ and p_{ji} is the transition probability from mode j to mode i over one time interval.

Using (2.3.3) and (2.3.4), one can rewrite (2.3.2) as

$$\Lambda_l(k) = \sum_{j=1}^{r} \left\{ \mathcal{N}\left[\mathbf{z}(k); \hat{\mathbf{z}}_l^i(k|k-1), S_l^i(k)\right] \sum_{j=1}^{r} p_{ji}\mu_l^j(k-1) \right\} \tag{2.3.5}$$

Then

$$l^* = \arg\max_l \Lambda_l^{k_0,K} = \arg\min_l \left[-\ln \Lambda_l^{k_0,K}\right] \tag{2.3.6}$$

selects the *best IMM estimator*. This yields the best initial estimate of the drag coefficient.

Based on the MIMM estimator, we can choose at the end of the observation period the most likely initial drag coefficient of the most probable mode from the best IMM estimator (2.3.6).[5] Then a numerical open-loop nonlinear predictor (we use the 4th order Runge-Kutta (RK) method [6] [20], incorporating the wind

[5] In practice, the thrusting period is relatively short and by the end of the observation period the BM should be the dominant one in the IMM. Otherwise, unless one were clairvoyantly to know the burnout time, one cannot make a meaningful IPP.

[6] Using the second-order open-loop EKF directly with a small iteration time interval (0.1 s)

effect if the wind information is available) is employed to predict the trajectory down to its impact point. At the same time, the corresponding covariance is also predicted to the impact point using a zero-gain (open loop) EKF covariance equation. Then the 99% chi-square probability region ellipse for the true impact point is [2]

$$[\, x_{\text{ip}} - \hat{x}_{\text{pd}} y_{\text{ip}} - \hat{y}_{\text{pd}} \,] P_{\text{pd}}^{-1} \begin{bmatrix} x_{\text{ip}} - \hat{x}_{\text{pd}} \\ \\ x_{\text{ip}} - \hat{y}_{\text{pd}} \end{bmatrix} = \chi_2^2 (99\%) \tag{2.3.7}$$

where $\begin{pmatrix} x_{\text{ip}} & y_{\text{ip}} \end{pmatrix}$ is the true impact point, $\begin{pmatrix} \hat{x}_{\text{pd}} & \hat{y}_{\text{pd}} \end{pmatrix}$ is the predicted impact point, P_{pd} is the corresponding predicted covariance matrix, and $\chi_2^2 (99\%)$ denotes the 99% point on the chi-square cumulative distribution function with two degrees of freedom [2]. This can be used as the "warning zone" in practical situations. In practical situations the ellipse is centered at the predicted impact point and one can check whether the true impact point (if available) is within the 99% probability ellipse [27]. In Monte Carlo simulations, as in present study, we evaluate the (equivalent) converse: whether the predicted impact point falls into the ellipse centered at the true impact point [21].

The IPP procedure based on the MIMM estimator is shown in Fig. 2.3.1.

2.4 MIMM Design Parameter Selection

The MIMM estimator using unbiased mixing with different initial drag coefficient estimates is used. The unbiased mixing is necessary because of the

for the prediction shows IPP performance practically the same as the RK method.

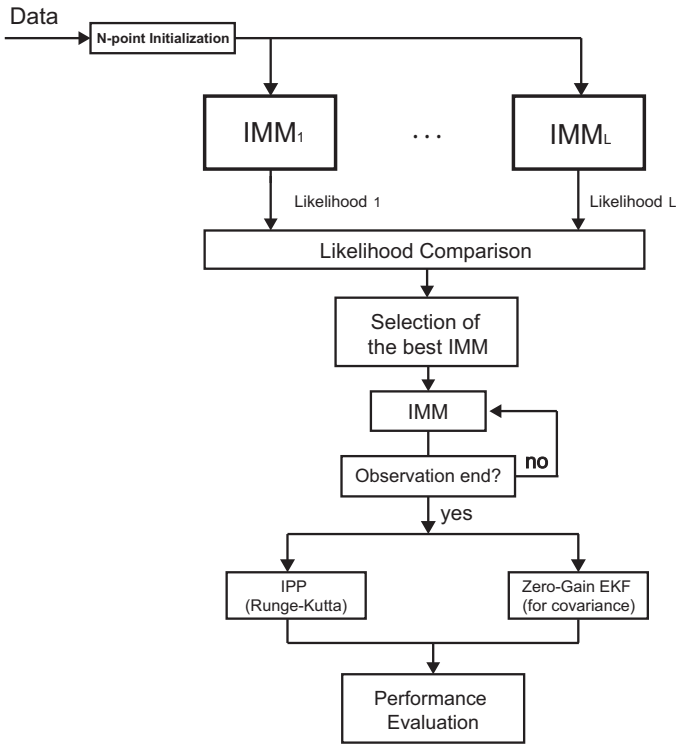


Figure 2.3.1: IPP procedure based on the MIMM-UM estimator.

unequal dimension state vectors in the two modes (TM and BM).

2.4.1 Selection of the Best IMM Estimator

The MIMM estimator is chosen to consist of $L = 4$ IMM estimators with initial drag coefficient estimates $\hat{\alpha}^i(0)$, $i = 1, 2, 3, 4$ as 0.18, 0.13, 0.065 and 0.03 (m^2/kg), respectively.[7] The initial SD of each drag coefficient estimate is 25% of the corresponding initial estimate of the drag coefficient.

If the projectile library with the truth of the corresponding drag coefficient information is provided, the selection of the best initial estimate of the drag coefficient actually indicates the projectile identification [8] [21]. However, the

[7]This grid of values was chosen based on the tracking results in the absence of wind. It seems that the 60 mm projectile has a somewhat different drag coefficient in the presence of wind (it turns somewhat slowly into the wind, perhaps the result of a larger moment of inertia).

wind effect could blur the identification.

2.4.2 N-point Adaptive Initialization

The initialization is crucial to the accuracy of the state estimates. Due to the sensitivity of the estimation on the drag coefficient, which is velocity (Mach number)-dependent and "marginally distinguishable" from the thrust estimate, an N-point adaptive initialization method is used here (the number N is discussed later). This method is based on the polynomial fitting of a set of noisy position measurements [2], with the polynomial order adaptively chosen based on the corresponding goodness-of-fit error. Particularly, it can give a good guideline for how to initialize the key thrust component.

The N-point fitting of a polynomial of order n is done as follows. The position measurements for the three Cartesian coordinates are

$$\begin{aligned}\mathbf{z}(k) &= H_{\mathbf{a}}(k)\mathbf{a} + w(k) \\ &\triangleq \begin{bmatrix} \mathbf{h}(k)' & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{h}(k)' & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{h}(k)' \end{bmatrix} \begin{bmatrix} \mathbf{a}_x \\ \mathbf{a}_y \\ \mathbf{a}_z \end{bmatrix} + w(k)\end{aligned} \tag{2.4.1}$$

where $k = 1, 2, \ldots, N$ and

$$\mathbf{h}(k) = \left[1 \;\; t_k \;\; \cdots \;\; \frac{(t_k)^n}{n!} \right]' \tag{2.4.2}$$

with t_k is the sampling time and the parameter vectors

$$\mathbf{a}_i = [\, a_{i0} \;\; a_{i1} \;\; \cdots \;\; a_{in} \,]' \qquad i = x,\; y,\; z \tag{2.4.3}$$

contain the coefficients of the polynomials (one for each coordinate).

For the time covered by N points, we get the estimate for the parameter vector $\mathbf{a}$ (of dimension $n_{\mathbf{a}} = 3(n+1)$) as

$$\hat{\mathbf{a}} = \left[H_{\mathbf{a}}^{N\prime} \left(R^N \right)^{-1} H_{\mathbf{a}}^{N} \right]^{-1} H_{\mathbf{a}}^{N\prime} \left(R^N \right)^{-1} \mathbf{z}^N \tag{2.4.4}$$

with the corresponding covariance matrix

$$P_{\mathbf{a}} = \left[H_{\mathbf{a}}^{N\prime} \left(R^N \right)^{-1} H_{\mathbf{a}}^{N} \right]^{-1} \tag{2.4.5}$$

where

$$\mathbf{z}^N = \begin{bmatrix} \mathbf{z}(1) \\ \vdots \\ \mathbf{z}(N) \end{bmatrix} \tag{2.4.6}$$

is the stacked vector of measurements (of dimension $3N \times 1$),

$$H_{\mathbf{a}}^{N} = \begin{bmatrix} H_{\mathbf{a}}(1) \\ \vdots \\ H_{\mathbf{a}}(N) \end{bmatrix} \tag{2.4.7}$$

is the stacked measurement matrix (of dimension $3N \times 3(n+1)$), and

$$R^N = \begin{bmatrix} R(1) \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots R(N) \end{bmatrix} \tag{2.4.8}$$

is the block diagonal covariance matrix of measurement noise (of dimension $3N \times 3N$) for the fitting interval.

The goodness-of-fit error has the following chi-square distribution [2]

$$J_N \stackrel{\Delta}{=} \left[\mathbf{z}^N - H_{\mathbf{a}}^N \hat{\mathbf{a}}\right]' \left(R^N\right)^{-1} \left[\mathbf{z}^N - H_{\mathbf{a}}^N \hat{\mathbf{a}}\right] \sim \chi^2_{3N-n_{\mathbf{a}}} \tag{2.4.9}$$

Using as acceptance region for (2.4.9) its 95% probability region (one-sided) and a component statistical significance test (a Gaussian test for the absolute value with 95% probability region, i.e., two-sided), we can adaptively choose the best order n^*, which avoids both "over-parameterizations" and "under-parameterizations" [2]. This is done by starting with $n = 1$ and increasing it until: (i) the test statistic (2.4.9) falls below the 95% probability threshold and (ii) the magnitude of at least one of the components (x, y and z) is statistically significant with the threshold $\mathcal{G}(97.5\%)$, which amounts to cutting the upper and lower tails of 2.5%.

The N-point adaptive initialization used $N = 12$ (about 1 s of data). As an illustrative example, Table 2.4.7 shows the results of the polynomial fitting of an observation sequence of the trajectory 60C7H7[8] with constant velocity (CV, $n = 1$), constant acceleration (CA, $n = 2$) and constant jerk (CJ, $n = 3$) models in one of the MC runs. For this particular case, the CA ($n^* = 2$) model is chosen: the fitting error is $J_N^* = 44.0 < \chi^2_{3N-n_{\mathbf{a}}}(95\%) = 50.7$ and the z-component estimate magnitude is statistically significant: $2.3 > \mathcal{G}(97.5\%) = 1.96$. The test statistics that yielded the choice $n^* = 2$ are in boldface.

[8] This stands for caliber 60 mm projectile in the presence of 7 m/s crosswind and 7 m/s headwind.

Remarks: Initialization of Thrust

In the present simulation study, the CV ($n = 1$) model and the CA ($n = 2$) model may be selected. For $n^* = 2$, the acceleration estimate is used (after subtracting the gravity acceleration and drag vector) to obtain the initial estimate of the thrust, $\hat{\tau}(0)$; for $n^* = 1$ (low thrust case), then $\hat{\tau}(0) = g$ with SD $g/4$.

Table 2.4.7: Fitting of CV, CA and CJ models for trajectory 60C7H7 (with $\sigma_r = 10\,\text{m}$ and $\sigma_\theta = \sigma_\epsilon = 5\,\text{mrad}$)

Model	CV ($n = 1$)			CA ($n = 2$)			CJ ($n = 3$)		
Component	x	y	z	x	y	z	x	y	z
$\hat{a}_{i0}$	0.9	-5.6	7.8	5.8	-17.2	-14.4	10.3	-20.9	-11.9
$\sqrt{P_{i0}}$	6.8	8.3	10.4	9.3	11.3	14.2	11.0	13.3	16.7
$\lvert\hat{a}_{i0}\rvert/\sqrt{P_{i0}}$	0.1	0.7	0.8	0.6	1.5	1.0	0.9	1.6	0.7
$\hat{a}_{i1}$	119.9	117.2	155.9	91.4	187.2	289.2	29.3	238.6	253.1
$\sqrt{P_{i1}}$	10.4	12.6	15.8	39.1	47.4	59.4	90.2	108.9	136.7
$\lvert\hat{a}_{i1}\rvert/\sqrt{P_{i1}}$	11.5	9.3	9.8	2.3	4.0	4.9	0.3	2.2	1.9
$\hat{a}_{i2}$				51.2	-127.4	-242.2	344.6	-369.4	-69.0
$\sqrt{P_{i2}}$				68.3	82.7	103.7	389.4	471.8	591.3
$\lvert\hat{a}_{i2}\rvert/\sqrt{P_{i2}}$				0.8	1.5	**2.3**	0.9	0.8	0.1
$\hat{a}_{i3}$							-531.9	437.7	-316.9
$\sqrt{P_{i3}}$							694.7	842.9	1055.6
$\lvert\hat{a}_{i3}\rvert/\sqrt{P_{i3}}$							0.8	0.5	0.3
J_N		*52.9*			**44.0**			43.3	
$\chi^2_{3N-n_{\mathbf{a}}}(95\%)$		52.0			50.7			49.5	

2.4.3 Selection of Process Noise Intensities

In the dynamic equations, we actually assume a nearly constant velocity model (continuous time white noise acceleration — CWNA [2]) for the kinematic components and a Wiener process with slow variation for the drag coefficient and thrust. In order to satisfy this model's assumptions, we choose small process noise "intensities" (power spectral densities — PSD) q_v, q_α, and q_τ as

follows.

The process noise induced root mean square (RMS) rate of change in velocity over an interval Δ is

$$d_v \triangleq \frac{\sqrt{q_v \Delta}}{\Delta} \quad [(\mathrm{m/s})/\mathrm{s}] \tag{2.4.1}$$

For the drag coefficient the rate of change is

$$d_\alpha \triangleq \frac{\sqrt{q_\alpha \Delta}}{\Delta} \quad [(\mathrm{m}^2/\mathrm{kg})/\mathrm{s}] \tag{2.4.2}$$

For the thrust the rate of change is

$$d_\tau \triangleq \frac{\sqrt{q_\tau \Delta}}{\Delta} \quad [(\mathrm{m/s}^2)/\mathrm{s}] \tag{2.4.3}$$

Then we have the following estimator design procedure for selection of the process noise PSD based on the corresponding RMS rates of change

$$q_v = d_v^2 \Delta \quad (\mathrm{m}^2/\mathrm{s}^3) \tag{2.4.4}$$

$$q_\alpha = d_\alpha^2 \Delta \quad (\mathrm{m}^4/(\mathrm{kg}^2 \cdot \mathrm{s})) \tag{2.4.5}$$

$$q_\tau = d_\tau^2 \Delta \quad (\mathrm{m}^2/\mathrm{s}^5) \tag{2.4.6}$$

The process noise intensities are chosen based on the process noise induced RMS change (in velocity/ drag coefficient/ thrust) over an interval of $\Delta = 1$ s as shown in Table 2.4.8. Note that d_v is chosen differently for the four initial drag coefficient estimates $\hat{\alpha}^i(0),\ i = 1, 2, 3, 4$.

Table 2.4.8: RMS change rate due to process noise

Filter	d_v [(m/s)/s]				d_α [(m^2/kg)/s]	d_τ [(m/s^2)/s]
	$\hat{\alpha}^1(0)$	$\hat{\alpha}^2(0)$	$\hat{\alpha}^3(0)$	$\hat{\alpha}^4(0)$		
IMM (TM)	3.5	2.5	2	1	$0.12\hat{\alpha}^i(0)$/s	$0.25\,\hat{\tau}(0)$/s
IMM (BM)	1.6	1.4	1.2	0.9	$0.10\hat{\alpha}^i(0)$/s	N/A

2.4.4 IMM Mode Transition Probability Matrix

The Markov chain probability transition matrix for the two-mode IMM is

$$\pi = \begin{bmatrix} p_{11} & 1-p_{11} \\ 1-p_{22} & p_{22} \end{bmatrix} \tag{2.4.1}$$

where the elements of the matrix are obtained based on the mean sojourn time (MST) [2], s_1 and s_2, for the TM ($i = 1$) and the BM ($i = 2$) respectively

$$p_{ii} = 1 - \frac{T}{s_i} \qquad i = 1, 2 \tag{2.4.2}$$

and $T = 0.1\,\text{s}$ is the (fixed) sampling interval. Setting the MST as $s_1 = 2\,\text{s}$ and $s_2 = 50\,\text{s}$ for TM and BM respectively, we get the mode transition probability matrix

$$\pi = \begin{bmatrix} 0.950 & 0.050 \\ 0.002 & 0.998 \end{bmatrix} \tag{2.4.3}$$

with initial mode probability vector $[\,0.90 \;\; 0.10\,]$.

2.5 Simulation Results

Three categories of thrusting/ballistic projectiles of different calibers are con-

sidered: 60 mm, 81 mm and 120 mm. In all, 21 trajectories were generated using [35][9] with a flat-Earth model with quadrant elevation 45° (i.e., aimpoint is NE). The thrusting/ballistic projectiles are actually rocket-assisted with initial velocity around 250 m/s (which varies across different calibers). For each category, there are trajectories with no wind present, labeled as "W0"; with 5 m/s crosswind only, labeled as "C5"; with 10 m/s crosswind only, labeled as "C10"; with 5 m/s crosswind and 5 m/s headwind, labeled as "C5H5"; with 7 m/s crosswind and 7 m/s headwind, labeled as "C7H7"; with 5 m/s crosswind and 5 m/s tailwind, labeled as "C5T5"; with 7 m/s crosswind and 7 m/s tailwind, labeled as "C7T7". Note that the headwind blows against the motion of the projectiles while the tailwind blows in the travel direction of the projectiles [37]. Measurements were obtained with no missed detections and no false alarms.

The parameters of the trajectories of the different caliber projectiles considered, namely, the projectile range to impact (R^t), the impact time (T^t_{ip}), the sampling interval (T), the maximum ground speed (V^t_{max}), the ground speed at impact point (V^t_{ip}) and the apogee altitude (H^t_{apg}), are summarized in Table 2.5.9. The launch point of each projectile was at the origin of coordinates. The sensor location was [5000 4000 0] m.

As shown in Table 2.5.10, four different sensor accuracy settings are used and labeled as *Case* 1 (good sensor accuracy), *Case* 2 (poor angle accuracy), *Case* 3

[9] This is a high fidelity simulation for rocket-assisted projectiles, i.e., with initial "catapulted" velocity. The trajectories were generated based on models we cannot control and do not know.

(poor range accuracy) and *Case* 4 (the best sensor accuracy).

Table 2.5.9: Parameters of the trajectories considered (different caliber sizes)

Caliber	R^t (m)	T^t_{ip} (s)	T (s)	V^t_{max} (m/s)	V^t_{ip} (m/s)	H^t_{apg} (m)
60 mm	3600 ± 100	30 ± 1	0.1	240 ± 3	155 ± 3	1085 ± 5
81 mm	5700 ± 150	37 ± 1	0.1	295 ± 3	200 ± 3	1680 ± 10
120 mm	7000 ± 250	41 ± 1	0.1	315 ± 5	230 ± 5	2055 ± 10

Table 2.5.10: Sensor accuracy settings

	σ_r	σ_θ	σ_ϵ
Case 1	10 m	5 mrad	5 mrad
Case 2	10 m	10 mrad	10 mrad
Case 3	25 m	5 mrad	5 mrad
Case 4	5 m	3 mrad	3 mrad

With 100 Monte Carlo (MC) runs for each sensor accuracy, we examine the IPP performance with given wind information (g.w.i.) and with no wind information provided (n.w.i.) for various *total observation time percentages of the whole trajectory* (denoted as "OT" for conciseness): 15%, 20%, 25%, 30%, 40% and 50%. In particular, under the same sensor accuracy and the same OT setting, in each MC run, the same random number set (but different in different MC runs, of course) has been used for different trajectories (W0/ C5/ C10/ C5H5/ C7H7/ C5T5/ C7T7) to examine how significantly the wind affects the IPP performance.

Three different situations concerning the knowledge of the wind information are considered: the "g.w.i." (perfect knowledge), the "n.w.i" (no knowledge) and the "e.w.i" (an inaccurate knowledge of 30% positive deterministic error,

i.e., $\mathbf{v}_w + \mathbf{v}_b \stackrel{\Delta}{=} \mathbf{v}_w + 0.3\mathbf{v}_w$, where $\mathbf{v}_b$ is a deterministic bias). Note that all the IPP results are *by default* obtained with "g.w.i." unless "n.w.i" or "e.w.i" is indicated. The indicating term "g.w.i." is omitted in Figures, Tables and discussions if there is no ambiguity.

A sample estimated trajectory, indicating the portions corresponding to the MIMM filtering and to the IPP, including the estimated burnout point (BoP) and the uncertainty ellipse centered at the true impact point, is presented in Fig. 2.5.1.

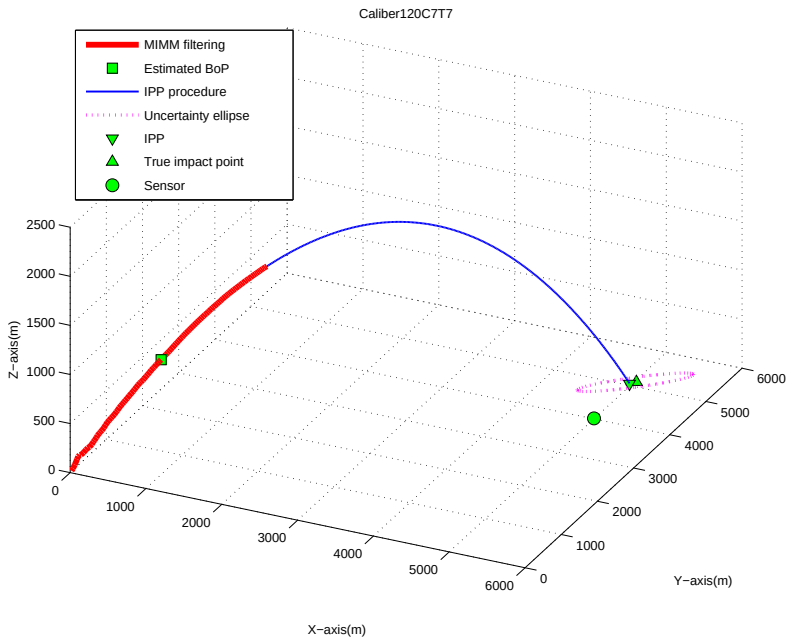


Figure 2.5.1: A sample estimated trajectory with IPP uncertainty ellipse centered at true impact point.

2.5.1 IPP RMS Error

The IPP root mean square (RMS) errors of the trajectories of caliber 60 mm, 81 mm and 120 mm (in various scenarios) are shown in Tables 2.5.11, 2.5.12 and

2.5.13, respectively. The g.w.i., n.w.i. and e.w.i. IPP RMS errors for various OTs and various sensor accuracies are evaluated and compared.

From Tables 2.5.11–2.5.13, it can be seen that, generally, as the wind becomes stronger, the difference between the g.w.i. IPP RMS error and the n.w.i. IPP RMS error will increase, especially for smaller OTs; as the OT increases, the influence of the wind on the IPP performance becomes smaller. As the sensor accuracy becomes worse, the IPP performance is gradually degraded. Particularly, a sensor error that causes more uncertainty along the travel direction of the projectiles, compared with the sensor error that causes more uncertainty perpendicular to travel direction of the projectiles, has more influence on the IPP performance. In the presence of a weak wind (C5), the IPP RMS error (g.w.i.) is practically the same as the IPP RMS error (e.w.i.). The knowledge of the wind information with a small deterministic bias has small effect on the IPP performance.

We can get an IPP RMS error of more than 500 m for the 60 mm projectile (Table 2.5.11: *Case* 2, OT=15%), more than 650 m for the 81 mm (Table 2.5.12: *Case* 3, OT=15%) and more than 750 m (even 900 m) for the 120 mm (Table 2.5.13: *Case* 2 and *Case* 3, OT=15%). These are unacceptable errors.[10] A strong wind effect, combined with a poor sensor accuracy, can lead to an unacceptable IPP RMS error when only a very short total observation time is available. On the

[10]In practical situations, a minimum acceptable IPP RMS error is expected to be within 10% of the projectile range. The IPP RMS errors larger than 10% of the projectile range are in *italic font* in Tables 2.5.11–2.5.13 .

other hand, a good sensor accuracy in the presence of known wind (even with a small bias) always yields a good IPP performance.

In Tables 2.5.11–2.5.13, the columns with the wind information provided (g.w.i.) show that, given a sufficient observation time (say, OT= 25%– 50%) to correctly select the best IMM from the MIMM estimator (i.e., to overcome the "marginal observability" problem between the drag coefficient and thrust), the IPP RMS errors are practically the same as without wind.[11] This implies that the wind effect can be fully compensated if the wind information is correctly provided.

Table 2.5.11: IPP RMS errors (in m, 100 MC runs) for various OTs (shown in the first column) and various sensor accuracies, caliber 60 mm: *italic*, see footnote 10; **bold**, see footnote 11.

Case 1	60W0		60C5			60C10			60C5H5			60C7H7			60C5T5			60C7T7	
		g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.
15%	417.4	419.5	420.7	419.8	441.4	426.7	440.1	452.5	461	459.6	458.4	464.3	465.3	417.2	426.8	418.5	408.8	409.6	410.9
20%	152.1	150.8	154.7	151.3	150.4	169	157.3	145.6	129.5	138.7	142.1	136.4	130.2	158.2	182	180.1	157.7	191.5	195.2
25%	127	127.1	132.8	127.5	126.9	145.2	125.8	122.6	111.8	97.4	120.7	119	91.2	127.7	142.2	158.6	120.2	155.9	166.5
30%	106	105.6	111.2	107.1	108.5	124.1	108.5	96	85.2	75.6	94.3	94.9	70.9	108.8	121.4	133.9	101.5	129.7	137.9
40%	60.4	60.2	65.3	59.9	59.5	78.8	60.2	59.2	55.4	46.3	59.9	59.1	45.4	60.3	72.3	80	59	82.9	88.2
50%	**40**	**40**	44.8	40	**39.3**	56.1	40.8	**40.2**	39.5	30.2	**40.5**	42.3	29.7	**40.2**	50.4	53.7	**39.4**	55.9	59.9
Case 2	60W0		60C5			60C10			60C5H5			60C7H7			60C5T5			60C7T7	
		g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.
15%	*537.7*	*533.3*	*539.4*	*534.5*	*532.3*	*537.3*	*536.5*	*535.6*	*535.4*	*544.8*	*538.6*	*546.1*	*552.6*	*530*	*523.7*	*516.8*	*505.1*	*519.4*	*508.1*
20%	209.1	207.1	207.3	207.3	203.7	218.2	206	197.9	194.4	190	197.4	192.6	185.4	199.1	216.4	224.7	205	216.6	236.9
25%	149.2	149.1	154.8	150.4	148.9	167.1	154.6	147.1	135.9	122.1	143.9	133.2	115.5	155.8	173.7	187.1	155.2	184	205.4
30%	114.3	114	119.6	114.2	114.5	130.9	116.3	111.7	104.3	89.7	111.2	104.4	84.1	119.3	138.2	147.4	117.4	145.3	160
40%	65.6	65.4	70.9	65.5	65.3	83.2	67	67	63.3	52.3	66.1	65.3	47.8	66.4	80.4	85.3	65.9	88.2	95.3
50%	**47.6**	**47.4**	52.6	47.2	**47.3**	64.9	47.9	**47.1**	46.3	35.9	**45.4**	49.8	32.6	**49.5**	59.3	62.4	**49**	65.1	69.2
Case 3	60W0		60C5			60C10			60C5H5			60C7H7			60C5T5			60C7T7	
		g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.
15%	373.2	372.5	376.2	374.3	369.4	383.5	391.7	379.1	347.4	378.8	380	336.3	381.7	363.7	385.5	384	361.1	393.4	392.6
20%	337.7	339.8	337.4	349.8	353.6	345	372.7	368	369.3	398.5	370.1	384.1	404.3	332.9	350	336.8	342.3	353	346
25%	175.7	176.2	179.7	178.3	176.6	193.2	182.2	180.8	170.4	168.3	174.3	168.7	167.6	182.6	207.4	203.8	181.4	219.2	213.2
30%	140.2	138.6	141.2	138.8	140.2	162	149.4	143.4	138.3	135.3	142.6	138.8	136.3	155.4	169.3	170.9	144.2	181.8	162.3
40%	110.7	111.4	115.8	115.2	116.2	126	113.8	103.4	98.6	98.2	104.6	104.4	97.6	115	122.8	127	113.6	122.1	128.9
50%	**54.9**	**55.4**	59.2	57	**55.8**	69.5	58.7	**63**	65.6	61.1	**62.2**	68.6	60.8	**57.5**	63.7	62.6	**58.4**	68	66.3
Case 4	60W0		60C5			60C10			60C5H5			60C7H7			60C5T5			60C7T7	
		g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.
15%	242.9	242.8	248.2	245	239.8	262	260.3	239	233.6	244.7	241.7	235.1	256.7	249.5	259.6	263.6	247.6	272.5	265.4
20%	137.4	139.4	143.1	140.8	138.3	159.6	147.7	137	132.8	116.5	136.6	133.3	116.8	140.6	166.4	178.3	140.6	173.6	189.3
25%	114.3	114	122.3	112	109.9	137.6	118.3	113.2	115.6	89.3	112.4	118.1	82.1	107.6	132.8	148.6	109.1	141.1	161.8
30%	88.7	89	96.7	90.1	88.3	112.2	96.2	89.4	88.1	71.1	89.1	91.9	70.2	86.5	102.7	118.3	84.5	109.9	127.4
40%	53.7	53.7	61.4	53.7	52.6	75	57.2	53.3	52.1	39.2	53	56.2	38.9	51.5	66.3	73.5	52.3	73.2	83
50%	**38.8**	**38.9**	42.5	38.6	**38.2**	52	40.7	**36.7**	36.4	26.6	**36.6**	39	25.6	**37.8**	46.8	52.2	**37.8**	52	58.3

[11] This is highlighted by the IPP RMS errors shown in **bold font** in Tables 2.5.11–2.5.13 .

Table 2.5.12: IPP RMS errors (in m, 100 MC runs) for various OTs (shown in the first column) and various sensor accuracies, caliber 81 mm: *italic*, see footnote 10; **bold**, see footnote 11.

Case 1	81W0		81C5			81C10			81C5H5			81C7H7			81C5T5			81C7T7	
		g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.
15%	420.8	420.1	424.1	422.2	420	438.7	427.4	401.8	423.3	428.6	389.3	436.2	399.1	421.1	382.8	414.3	420.3	384.9	415
20%	432.4	428.4	438.6	429	428	445.9	430.9	423.4	401.8	384.1	418.6	393.2	361.2	428.7	456.5	481.9	430.1	475.5	497.6
25%	288.9	294.4	294.5	292.3	292.7	300.6	296.3	292.3	260.8	254.7	291.1	257.4	240.3	282.7	323.6	333	285.2	343.8	352.6
30%	188	186	193.3	186.8	185	204.1	190	190.2	181.2	158	196.6	183.3	148.5	189.3	206	230.1	191.5	221.3	246
40%	101.1	100.5	104.8	101	100.8	118.5	103.4	99.4	96.7	80.4	99	96.8	72.7	99.3	114.2	127.1	101	122.4	141.6
50%	**48.7**	**48.8**	51.8	48.6	**49.7**	62.2	51.6	**48.5**	47.9	40.1	**47.7**	49.7	36.8	**48.5**	58.6	63.7	**49.5**	64.5	71.9
Case 2	81W0		81C5			81C10			81C5H5			81C7H7			81C5T5			81C7T7	
		g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.
15%	497.5	492	492.2	491.4	482.9	501.6	484.2	490.5	520.3	500.4	511.8	504.4	528.4	491.9	526.6	494.1	486.7	534.8	500.7
20%	450.5	449.6	451.7	448.9	448.9	459.7	448.8	451.7	416.9	398.2	441.6	401.2	383	444.3	467.6	486.7	449.9	495.5	510.2
25%	315.7	313.6	320	317	313.1	328.4	318.4	318.8	295.6	283	317.8	285.9	269.3	317.1	354.3	359.6	320	376.9	377.2
30%	209.5	209	217.1	209.8	208.4	230.2	211.6	209.3	204.6	175.3	217.2	214.8	174.7	212.5	239.6	251.8	213.7	257.8	267.3
40%	133.8	135.2	140	135.4	134.9	151.3	137.1	135.3	130	111.5	134.9	130.9	103.3	131	150.9	157.7	129.3	159.9	166.4
50%	**68.9**	**69.1**	73	68.6	**72.7**	87.5	73.6	**72.1**	69.7	57.9	**71.8**	71.7	53.1	**69.1**	80.1	86.1	**69.6**	87.7	94.7
Case 3	81W0		81C5			81C10			81C5H5			81C7H7			81C5T5			81C7T7	
		g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.
15%	*665.2*	*665*	*667.1*	*665.9*	*652.6*	*675.7*	*659.2*	*664.8*	*661.7*	*643.9*	*690.8*	*703.7*	*668.2*	*646.5*	*673.5*	*667.4*	*651.7*	*687.6*	*667.4*
20%	479.9	479.6	480.9	479.7	485	490.7	490.9	478.3	473.4	456	472.8	471.7	445.3	476.8	473.2	504.8	476.6	501.9	516.7
25%	344.9	345.1	349.3	346.4	348.2	362.3	352.3	351.4	327.9	331	348.3	320.9	325.5	347.4	375.1	365.2	346.9	372.1	376
30%	213.5	213.7	216.9	212	217.5	229.5	218.2	212.7	201.5	190.4	223.4	203.3	208.9	217.9	240.9	238.4	219.1	248.8	252.2
40%	131.2	127.2	136.6	126.9	126.9	144.4	131.7	131.4	130.7	124.4	128.4	134.1	129.7	131.6	145.5	139.4	129.9	143	145.2
50%	**66.7**	**67**	70.4	67	**63.9**	74.8	67.1	**62.5**	63	60.4	**64**	65.3	60	**66.8**	71.1	76.4	**67.7**	77.8	83
Case 4	81W0		81C5			81C10			81C5H5			81C7H7			81C5T5			81C7T7	
		g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.
15%	465.4	465	468.2	465.2	464.9	479.5	467.3	464.9	444.7	395.7	445.1	451.4	375.8	464.7	469.2	520.1	463	483	549.7
20%	322.4	322	332.9	322.3	321.5	342.7	323.6	324.1	294.5	268	323.2	286.4	256.3	317.9	362.1	368.7	301.1	367.5	375.8
25%	199.8	202.1	208.2	203.1	202	223.4	206.9	200.4	176.8	165	198.5	174.7	154	201.8	235	242.5	205.2	252.5	261.4
30%	123.9	122.4	129.1	123.2	121.3	145.6	130.2	121.1	120.6	95.8	124.3	138	94.3	122.6	140.2	166.6	121.2	151.4	184.8
40%	67.7	67.7	73.5	68.8	67.7	85.2	72.6	67.4	65.6	52.7	67.5	68.2	50.9	67.8	80.8	91.8	65.7	87.7	101.6
50%	**33.5**	**33.2**	37.3	34.7	**36.2**	48.6	40.8	**36.2**	35.5	28.1	**35.9**	37.6	28.3	**33.8**	42.7	49.2	**34**	48.2	58.1

From Tables 2.5.11–2.5.13, an interesting observation is that for the situations with headwind and very short observation time, e.g., C5H5 and C7H7 with small OTs (e.g., OT=15% or 20%), the g.w.i. IPP RMS performance is counter-intuitively worse than the n.w.i. IPP RMS performance. This is because of the imperfect selection of which is the best IMM in the MIMM estimator when it is based on a very short observation time. More clearly, as shown in Fig. 2.5.2 (the trajectory 60C7H7 in *Case* 3 and OT=15%), because of this imperfect decision of the best IMM, each MC IPP "cloud of points" (for g.w.i. and n.w.i.) is approximately separated into 3 different clusters, designated as the "cluster

Table 2.5.13: IPP RMS errors (in m, 100 MC runs) for various OTs (shown in the first column) and various sensor accuracies, caliber 120 mm: *italic*, see footnote 10; **bold**, see footnote 11.

Case 1	120W0		120C5			120C10			120C5H5			120C7H7			120C5T5			120C7T7	
		g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.
15%	718.2	718.2	720	718.1	717.8	714.7	718.8	749.5	653.1	714.2	*753.2*	652.5	698.9	684.6	*760.6*	723.1	681.3	779.5	745.6
20%	547.4	546.7	548.5	547.4	539.6	555.6	544.5	576.7	562.3	534.9	572.4	575.6	477.1	581.2	571.5	651.9	591.7	585.5	699.5
25%	488.9	492.6	492	492.9	491.9	497.4	497.9	488.5	474	416.3	484.2	465.9	391.2	502.9	521.7	552.4	494.8	513.7	578
30%	308.6	308.4	315.5	308.4	307.8	330.4	308.9	306.4	292.7	261.8	302	284.7	243.5	313	334	354.3	316	337.8	368.4
40%	117.5	115	122.6	115.8	114.8	132.6	117	114.6	116.6	100.4	115.4	118	94.9	121.5	130.1	142.2	118.2	128.7	150.7
50%	**55.1**	**54.9**	58.8	56.6	**56**	67.2	59.8	**52.4**	55.4	52.9	**51.4**	57.5	54	**56**	60.6	62.6	**56.7**	64.1	66.8
Case 2	120W0		120C5			120C10			120C5H5			120C7H7			120C5T5			120C7T7	
		g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.
15%	*779.2*	*776.3*	*790.9*	*775.8*	*782.2*	*789*	*777.1*	*769.3*	748.9	*766.6*	*786*	725.8	*756.8*	*761.2*	*797.5*	*773.3*	744.9	*788.3*	*795.7*
20%	613.3	613.3	618	612.3	616	622.6	614.8	597.5	593.1	535.7	607.2	571.8	509.9	624.1	631.1	693.5	621.4	653.9	725
25%	489.8	494.1	492.5	493.5	493.6	497.9	492.9	493.2	480.9	450	497.5	473.7	419.3	493.9	500.7	557.1	475.7	486.2	551.6
30%	346.7	348.2	349.8	347.7	347.9	354.3	345.2	362	346.4	317.5	359.7	344.5	299.6	345.9	368.2	387.8	342	381.2	414.3
40%	144.5	142.9	148.8	143.5	142.3	158.3	144.4	144.9	148.7	126.7	144.9	149.1	119.6	152.3	159.1	179.4	156.4	174.1	197.2
50%	**63**	**61.5**	66.7	62	**61.1**	75.7	63.4	**61.6**	63.4	53.7	**59.8**	64.8	50.4	**64.7**	69.1	79.6	**63.1**	73.8	84.9
Case 3	120W0		120C5			120C10			120C5H5			120C7H7			120C5T5			120C7T7	
		g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.
15%	*887.6*	*879.9*	*885.9*	*879.6*	*879.2*	*889*	*879.3*	*874.7*	*859.6*	*833.7*	*870.8*	*867.2*	*817.1*	*883.2*	*894.5*	*918.5*	*878.7*	*901.2*	*936.4*
20%	650.1	651.2	654.4	652.9	648.6	659.1	643.8	634.1	611.4	612.4	642.2	608.4	607.2	639.8	686.1	673.8	642.7	697.9	685.1
25%	446.3	446.4	443.7	433.6	431.7	445	431.3	449.1	397.7	433.1	437.6	395.9	425.4	426.6	465.4	450.9	398.9	422.2	452.6
30%	279.3	271.4	281.5	272.5	270.1	302.4	275.9	284.4	277.7	273.1	285.7	275.8	265.9	272.5	284.1	294.4	272.4	286.6	314.3
40%	130.5	130.8	135.6	131	127.1	146.5	133	136.7	135.4	132.3	134.3	138.6	131.7	129.1	139.5	132.4	133.3	151.4	144
50%	**64.9**	**65.1**	68.1	67.7	**64.4**	75	71	**64.3**	67.3	65.5	**64.1**	70.2	66.6	**66.7**	70.3	68.1	**62.6**	68.1	71.5
Case 4	120W0		120C5			120C10			120C5H5			120C7H7			120C5T5			120C7T7	
		g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.	g.w.i.	n.w.i.	e.w.i.
15%	462.7	462.4	462.2	477.3	476.6	468.3	482.2	480.2	445.6	426.4	464.8	456.8	453.1	464.1	534.3	533.8	474.7	528.1	590.9
20%	433.9	433.3	433.8	434.1	430.2	448.5	434.6	413.3	420.6	350.8	413.8	445.6	306.8	444	430.8	506	456	437.5	543.7
25%	337.4	336.6	340.7	336.9	335.5	353.1	339.8	323.9	318.7	276	321.8	321.2	260	350.1	360.3	394.8	329.5	368.2	398.2
30%	198.1	198	206.4	196.7	196	213.1	200.7	177.9	170.4	151.8	178.9	172.2	143.1	207.9	221.4	246.6	211	235.1	268.4
40%	62.2	62	65.1	62.9	61.4	76	67.6	61.8	65.8	59.7	59.9	69.2	64.2	64.5	68.9	80.2	67.7	77.8	93
50%	**29.5**	**29.4**	32.4	30.9	**29.4**	40.7	35	**30.5**	33	32.5	**29.8**	36.2	34.9	**30.2**	34.6	38.6	**29.7**	37.9	43.2

A", "cluster B" and "cluster C". It can be seen, compared with the g.w.i. clusters, that the n.w.i. clusters have a clear drift caused by the wind effect. By coincidence, this drift happens to drive "cluster A (n.w.i.)" that contains the most points of the MC IPP closer to the true impact point. This results in the g.w.i. IPP RMS errors being worse than the corresponding n.w.i. IPP RMS errors in some cases in Tables 2.5.11–2.5.13. However, as the observation time increases, the MIMM estimator can gradually make the correct decision to find the best IMM estimator (the one with the most suitable initial drag coefficient estimate) and then, as we expect, the g.w.i. IPP results are better than the

corresponding n.w.i. IPP results. Given a very short OT in a strong wind environment (e.w.i. vs. g.w.i.), the bias in the erroneous wind information sometimes exacerbates and sometimes improves the IPP results. Generally, the e.w.i. IPP performance is between the g.w.i. and n.w.i. cases when the MIMM-selected drag coefficient estimate matches the truth.

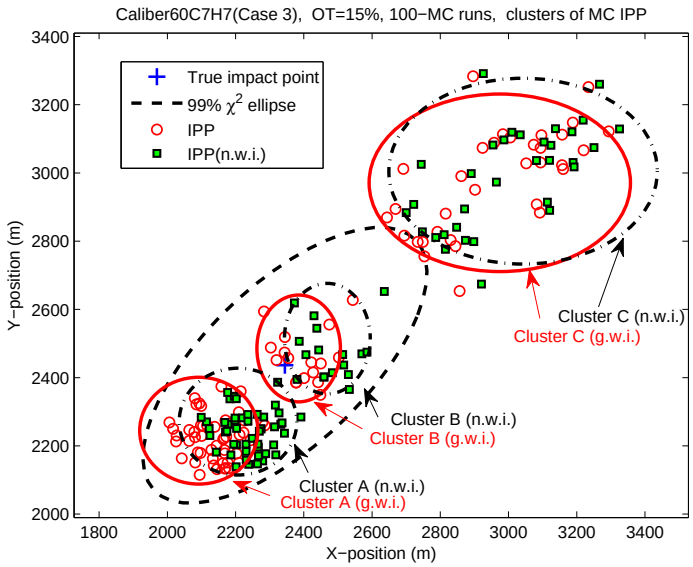


Figure 2.5.2: Clusters of MC IPP "clould of points" caused by imperfect decision of the best IMM in MIMM estimator.

2.5.2 IPP Uncertainty Ellipse

With various OTs and sensor accuracies, the IPP uncertainty ellipses using (2.3.7), centered at the true impact point, with the corresponding *MC IPP "cloud of points"* (for both g.w.i. and n.w.i.) for various trajectories in the presence of wind are shown in Figs. 2.5.3–2.5.5.

From Fig. 2.5.3, it can be seen that as the OT increases, more IPP points fall in the corresponding uncertainty ellipses and the sizes of the uncertainly ellipses

gradually shrink. The total observation time is crucial for the evaluation: given the OT is large enough (50%), most of the MC IPP points fall within the uncertainty ellipses with acceptable IPP RMS errors. The outliers are due to the forced early decision when the OT is small.

From Fig. 2.5.4, it can be seen that as the sensor accuracy worsens, the corresponding IPP uncertainty ellipses become wider because of the poor angle accuracy or become longer because of the poor range accuracy, and fewer MC IPP points fall into them. Particularly, the larger sensor errors in angles (which cause more uncertainty along the minor axis of the uncertainty ellipse in the case considered) result in the MC IPP points being more scattered along the minor axis of the corresponding IPP uncertainty ellipse; the larger sensor errors in range (which cause more uncertainty along the major axis of the uncertainty ellipse in the case considered) result in the MC IPP points being more scattered along the major axis of the corresponding IPP uncertainty ellipse. Overall, the sensor errors along the projectile travel direction have more impact on the IPP performance.

From Fig. 2.5.5, comparing the n.w.i. IPP clouds with *smaller wind* ("C5"/"C5H5"/"C5T5") to those n.w.i. IPP clouds with *stronger wind* ("C10"/"C7H7"/"C7T7"), respectively, it is obvious that the stronger the wind is, the more significant the deviations of the n.w.i. IPP clouds are. When both the cross wind and the range wind were present, the wind-induced drift of the n.w.i. IPP clouds is compounded: without accounting for the cross wind, the n.w.i. IPP cloud

deviates to one side (the right side in the cases discussed) of the major axis of the uncertainty ellipses (obtained by properly accounting for the wind effect in the dynamic model) while the lack of accounting for the range wind causes the n.w.i. IPP cloud to be "pushed backward/forward" causing range under-/over-prediction.

2.5.3 Consistency Evaluation

The consistency of the MIMM estimator is very important in the IPP application. It is always desired that the predicted impact point falls into the uncertainty ellipse centered at the true impact point (or the equivalent converse). This can be achieved by increasing the process noise intensities, which results in an ellipse with its major- and minor- axes large enough to include the predicted impact point. However, this solution is undesirable. A consistency test will help to find the process noise setting that gives the uncertainty ellipse compatible with the actual errors [2].

The consistency of the MIMM estimator is examined using both the normalized estimation error squared (NEES), which is preferable for Monte Carlo runs when the truth is available (off-line simulations), and the normalized innovation squared (NIS). The latter is the only one that can be used in real time testing [2].

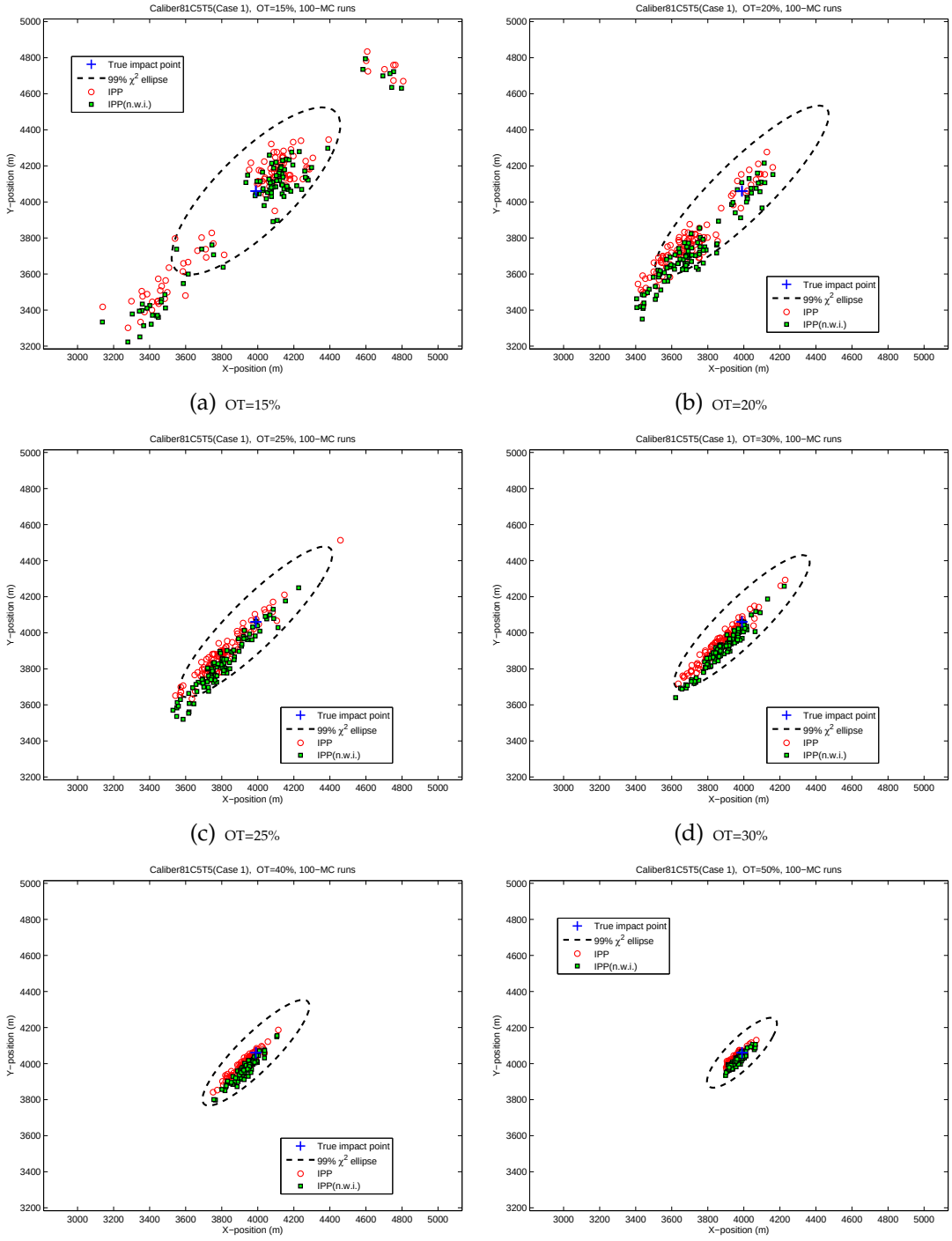
Fig. 2.5.6 shows, for the best IMM estimator selected from the MIMM estimator with OT=50% and various sensor accuracies, the NEES consistency in position

and velocity and the NIS. Note that the NIS is evaluated for TM before the *estimated* BoP and for BM afterward [27]. It can be seen that as the sensor errors increase, the IMM estimator selected becomes less consistent. In particular, the larger the sensor's range error (which causes more uncertainty along the travel direction of the projectiles in present cases), the more significant is the inconsistency.

2.6 Summary and Conclusions

The MIMM estimator developed earlier has been extended to account for the wind effect. The IPP performance for various total observation times and sensor accuracies have been investigated in detail. The wind effect can be fully compensated if the wind information is available; it also can be mitigated by increasing the total observation time if no wind information is provided. The total observation time is key for IPP performance: for example, given the observations all the way up to the apogee one can always expect a very good IPP performance, but this is clearly not desirable if countermeasures are to be taken against the projectile. The N-point initialization, designed for initializing the key thrust component as well as the kinematic components, is considered a good method to quicken the correct mode convergence in each IMM estimator and alleviate the estimation ambiguity between the drag coefficient and the thrust components. In the cases studied here, an OT of 20% or more and sensor accuracies under 10 m in range and 5 mrad in angles are required to overcome

the marginal observability (ambiguity) of the drag and thrust and then achieve an acceptable IPP performance, with the IPP RMS error no more than 10% of the projectile range. The sensor errors that cause more uncertainty along the travel direction of the projectiles have more impact on the IPP performance. In the presence of wind, the short observation time and the limited sensor accuracy are more critical since they could cause confusion in selecting the "correct" best IMM estimator (which should be the one with the most suitable initial drag coefficient estimate, closest to truth) from the MIMM estimator, and then lead to a degraded IPP performance.



(a) OT=15%

(b) OT=20%

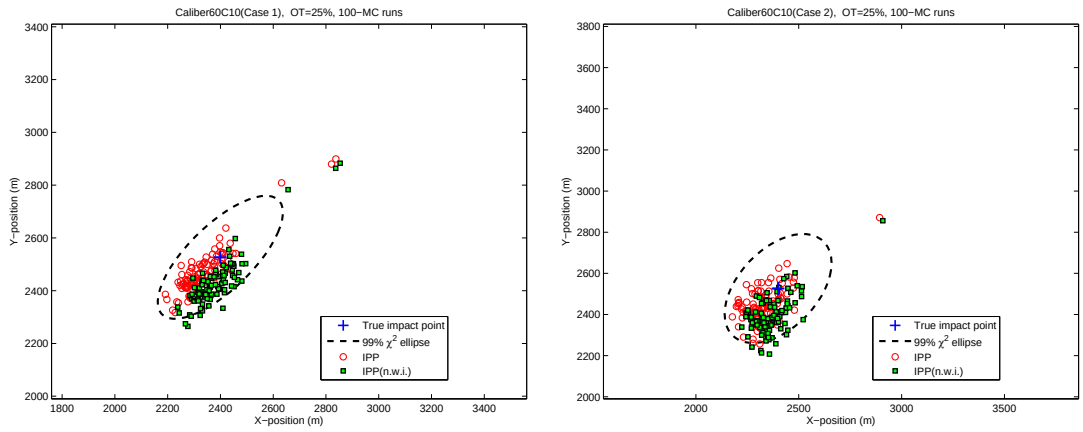
(c) OT=25%

(d) OT=30%

(e) OT=40%

(f) OT=50%

Figure 2.5.3: IPP uncertainty ellipse and MC IPP cloud for *various OTs*, trajectory 81C5T5, *Case* 1, 100 MC runs



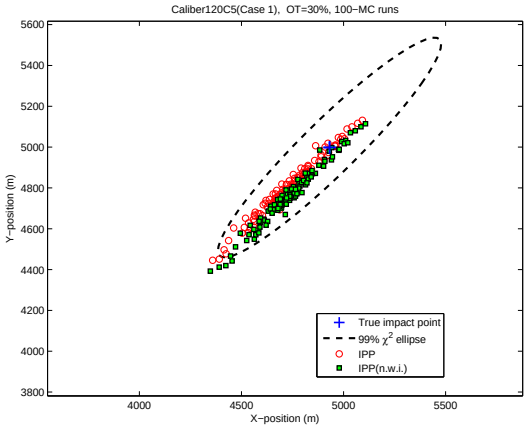
(a) *Case*-1 (good accuracy)

(b) *Case*-2 (poor angle accuracies)

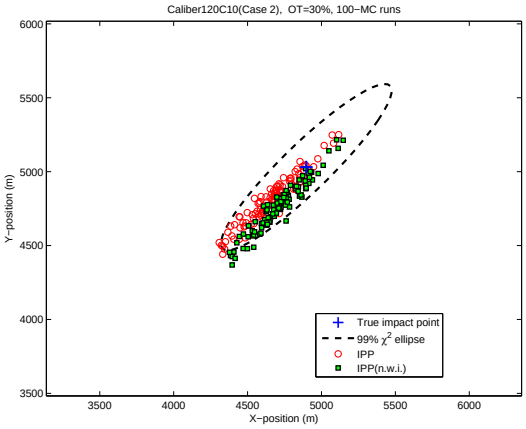
(c) *Case*-3 (poor range accuracy)

(d) *Case*-4 (best accuracy)

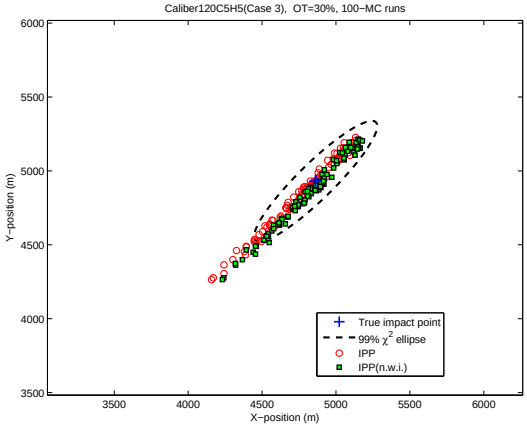
Figure 2.5.4: IPP uncertainty ellipse and MC IPP cloud for *various sensor accuracies*, trajectory 60C10, OT=25%, 100 MC runs



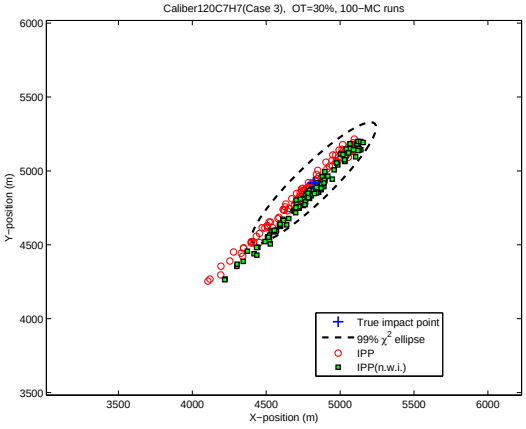
(a) Trajectory 120C5



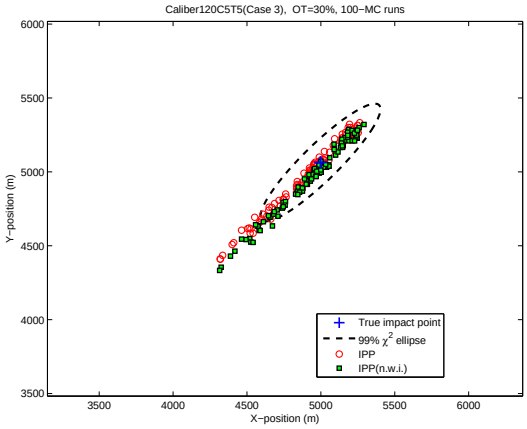
(b) Trajectory 120C10



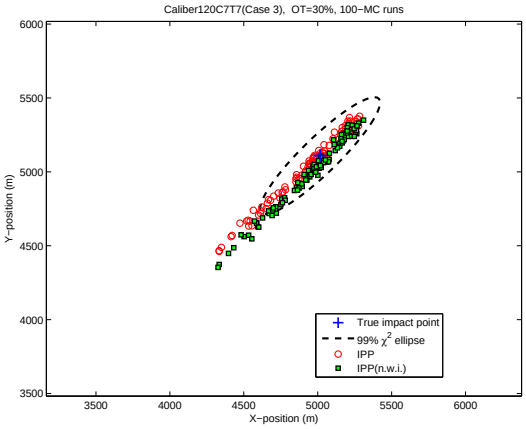
(c) Trajectory 120C5H5



(d) Trajectory 120C7H7

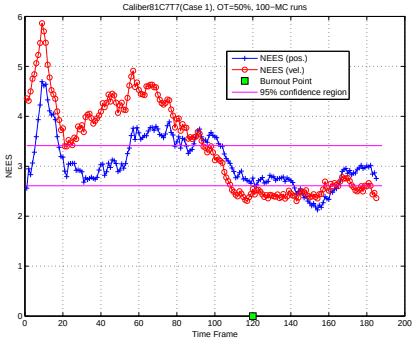


(e) Trajectory 120C5T5

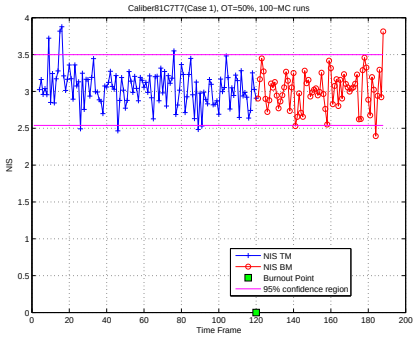


(f) Trajectory 120C7T7

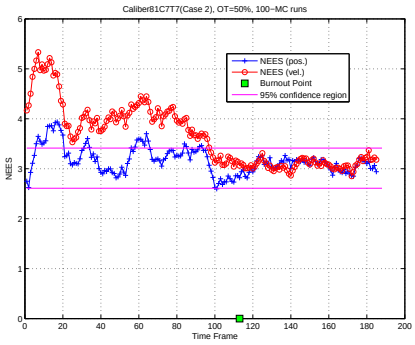
Figure 2.5.5: IPP uncertainty ellipse and MC IPP cloud for various 120mm trajectories, OT=30% 100 MC runs



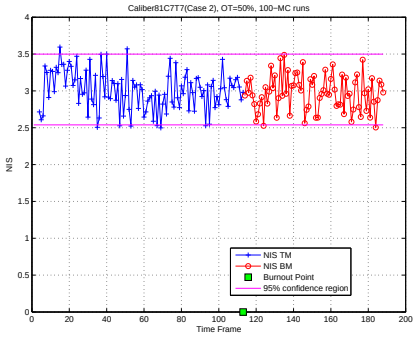
(a) NEES (pos. and vel.), *Case* 1



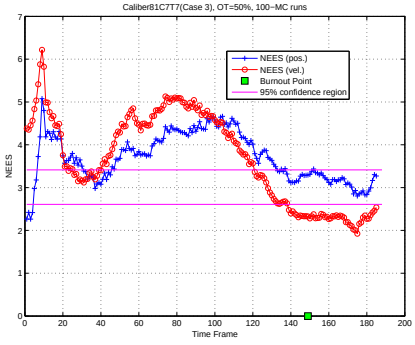
(b) NIS *Case* 1



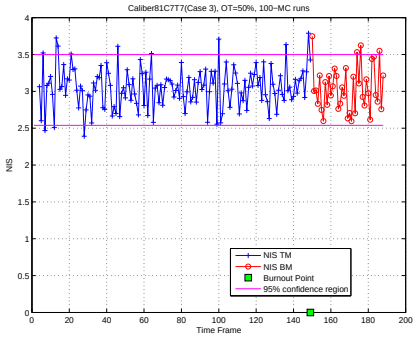
(c) NEES (pos. and vel.), *Case* 2



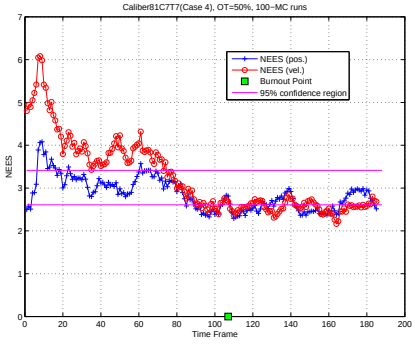
(d) NIS, *Case* 2



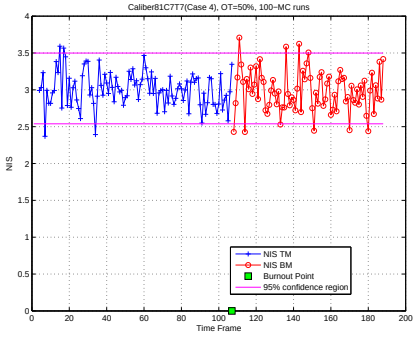
(e) NEES (pos. and vel.), *Case* 3



(f) NIS, *Case* 3



(g) NEES (pos. and vel.), *Case* 4



(h) NIS, *Case* 4

Figure 2.5.6: Consistency test (NEES in pos. and vel. and NIS, g.w.i.), trajectory 81C7T7, OT=50%, 100-MC run

Chapter 3

Estimation of Thrusting Trajectories in 3-D from a Single Fixed Passive Sensor

The problem of estimating the state of thrusting/ballistic endoatmospheric projectiles moving in 3-dimensional (3-D) space for the purpose of impact point prediction (IPP) using 2-dimensional (2-D) measurements from a single passive sensor (stationary or moving with constant velocity) is investigated. The location of projectile's launch point (LP) is generally unavailable and this could significantly affect the performance of the estimation and the IPP. However, if the altitude of the LP is known, the launch position can be obtained with negligible error from the first line of sight (LoS) measurement intersected with the terrain map. The estimability is analyzed based on the Fisher Information Matrix (FIM) of the target parameter vector that determines its trajectory: the initial launch (azimuth and elevation) angles, drag coefficient and thrust. The lack of knowledge about the LP altitude makes the problem substantially more difficult. The LP altitude is then an additional unknown target parameter and has to be included into the target parameter vector that needs estimability analysis. The full rank of the FIM, with/without given the LP altitude, ensures that one has estimable target parameters. The corresponding Cramér-Rao lower

bound (CRLB) quantifies the estimation performance of the estimator that is statistically efficient and can be used for the IPP accuracy evaluation. In view of the inherent nonlinearity of the problem, the maximum likelihood (ML) estimate of the target parameter vector can be found by using a suitable numerical approach. A search strategy with two stages — a mixed (partially grid-based) search followed by a continuous search — is proposed. For even a coarse grid, this approach is shown to have reliable estimation performance and leads to a final IPP of good accuracy, which is the ultimate goal. Due to its parallelizable nature, the mixed search allows the two-stage strategy to be real-time implementable.

3.1 Introduction

The estimation of the state of thrusting/ballistic endo-atmospheric projectiles moving in 3-dimensional (3-D) space for the purpose of impact point prediction (IPP) using 2-dimensional (2-D) measurements from a single passive sensor (stationary or moving with constant velocity) is investigated. The trajectory estimation problem has two major challenges. The first is that to estimate and predict the 3-D trajectory using the (incomplete) 2-D line of sight (LoS, i.e., azimuth and elevation angles) measurements instead of full position measurements [8][34], could exhibit a geometry-dependent ambiguity due to marginal observability. This requires an estimability analysis for the target parameters that determine the trajectory: the initial launch (azimuth and elevation) angles,

drag coefficient and thrust. The second is that the lack of knowledge about the location of the projectile's launch point (LP) exacerbates the estimation ambiguity between the drag coefficient and thrust estimands [34]. However, the altitude of the LP can be available from terrain information. The launch position is then obtained from the first LoS measurement intersected with the terrain elevation database (or, in our simplified case, the known-altitude plane). If the terrain elevation (altitude) information is unavailable, the estimation problem becomes substantially more difficult. The LP altitude is then an additional unknown target parameter. The burnout time (BoT) is assumed available from the passive (optical) sensor.

Estimability analysis for such a nonlinear system is a challenging task even with known LP altitude. In the literature the Fisher information matrix (FIM) has been used as a convenient tool in target motion analysis problems [10][26] to analyze estimability. The major difficulty of obtaining the estimability criterion in the present problem lies in the fact that the FIM can only be obtained numerically rather than analytically due to the nonlinear motion equations which require numerical integration. The corresponding Cramér-Rao lower bound (CRLB) provides (if the estimator is efficient) the accuracy of the parameter estimate, which can be mapped into IPP accuracy.

Following description of the target and observation models, we investigate the estimability of the target parameters assuming deterministic target motion during the observation interval by checking the invertibility of the FIM. A max-

imum likelihood (ML) estimator for the target parameters is developed. If the FIM is full-rank, the ML estimator will ideally be the global maximum of the likelihood function (LF) of the target parameters. Obstacles are that some necessary derivatives can only be obtained numerically, and the LF may have local maxima. A numerical search strategy with two stages is proposed. The first is a mixed search using a selected (discrete) drag-thrust[12] grid ("drag-thrust-altitude" grid is used instead, if the LP altitude is unknown) and continuous over the (2-D) launch angles space, followed by the second stage, continuous over the whole target parameter space in a contracted re-centered region.

The grid is crucial for a quick and correct convergence of the mixed search and the continuous search that follows. A fine grid yields good estimation and IPP performance [29][30] but leads to a high computational burden. In the present discussion, for a coarse grid (with its grid points far away from the corresponding truth), a simulation study is conducted to show that the proposed search strategy converges to the unique global maximum of the LF at each grid point and lead to the resulting ML estimate being mapped into the IPP of good accuracy. The parallelizable nature of the mixed search allows the two-stage strategy to be real-time implementable.

Motion of the passive sensor to enhance the system estimability is also investigated. Due to the ineluctable numerical integration, the improvement of a moving platform (MP) over the corresponding stationary platform (SP) could

[12]For simplicity, "drag-coefficient-thrust" will be called "drag-thrust" if there is no ambiguity.

not be shown analytically. However, with the aid of the CRLB it can be seen that the MP does accrue some minor benefit over the corresponding SP.

The chapter is organized as follows. The problem of estimating the 3-D trajectory using the measurements from a single passive sensor is presented in Section 3.2. In Section 3.3, the possible existence of a geometry-dependent ambiguity due to marginal observability is discussed and the ML estimator of the target parameter vector is developed. In Section 3.4, under assumption of known/unknown LP altitude the FIM, the CRLB and the search strategy with two stages are presented. In Section 3.5, simulations are presented based on the dynamic model described in [34]. Conclusions are presented in Section 3.6.

3.2 The Problem

Let the *target parameter vector* be

$$\Theta = [\ \vartheta_0 \ \ \varphi_0 \ \ \alpha \ \ \tau\ ]' \tag{3.2.1}$$

where the launch (initial) azimuth angle ϑ_0 and elevation angle φ_0 are used to specify the unknown launch direction of the projectile. The unknown drag coefficient α is assumed to be constant during the observation period. The unknown thrust τ is also assumed constant up to the BoT, the time of which is assumed for obvious reasons to be observable from the imaging sensor. Up to the BoT the projectile is in thrusting mode, after that it is in ballistic mode.

The projectile LP location with the exact knowledge of its full position is gen-

erally unavailable. However, the LP altitude can be available from terrain information and the projectile launch position is then estimated from the first LoS measurement intersected with the known-altitude plane (or the 3-D terrain elevation map[13]). With no such knowledge, the LP altitude h is an additional unknown target parameter and has to be included into the *augmented target parameter vector*

$$\bar{\Theta} = [\, \vartheta_0 \;\; \varphi_0 \;\; \alpha \;\; \tau \;\; h \,]' \tag{3.2.2}$$

In the following discussion, we will first focus on the known LP altitude assumption. The problem with the unknown LP altitude is handled in a similar manner with the augmented target parameter vector, as discussed in Section 3.4.

Under the known LP altitude assumption, the target state vector (consisting of position and velocity in Cartesian coordinates) for the motion of the projectile is

$$\mathbf{y}(t,\Theta) = [\, x(t,\Theta) \;\; y(t,\Theta) \;\; z(t,\Theta) \;\; \dot{x}(t,\Theta) \;\; \dot{y}(t,\Theta) \;\; \dot{z}(t,\Theta) \,]' \tag{3.2.3}$$

This is different from the stochastic state $\mathbf{x}$ used in recursive estimation [34], consisting of $\mathbf{y}$, α and τ, which evolve in the presence of process noise. Here we use a parameter estimation approach (assuming deterministic evolution of

[13]Clearly, this does not give a perfect LP, but the error is practically negligible.

$\mathbf{y}$, i.e., no process noise[14]) to determine the estimability of the target parameter vector from incomplete position observations given by a single passive sensor, i.e., only the azimuth and elevation angles. The time and parameter arguments t (later replaced by k for discrete time) and Θ (or $\bar{\Theta}$) will be omitted where no ambiguity ensues.

Based on the target parameters in (3.2.1), we have thrust components in Cartesian coordinates at time t

$$\tau_x(t) = \tau \cos \vartheta(t) \cos \varphi(t) \tag{3.2.4}$$

$$\tau_y(t) = \tau \sin \vartheta(t) \cos \varphi(t) \tag{3.2.5}$$

$$\tau_z(t) = \tau \sin \varphi(t) \tag{3.2.6}$$

where

$$\vartheta(t) = \tan^{-1}\left[\frac{\dot{y}(t,\Theta)}{\dot{x}(t,\Theta)}\right] \tag{3.2.7}$$

$$\varphi(t) = \tan^{-1}\left[\frac{\dot{z}(t,\Theta)}{\sqrt{\dot{x}^2(t,\Theta) + \dot{y}^2(t,\Theta)}}\right] \tag{3.2.8}$$

with initial condition

$$\vartheta(0) = \vartheta_0 \qquad \varphi(0) = \varphi_0 \tag{3.2.9}$$

Assuming a (stationary or moving) passive sensor is located at $[\, x_o(k) \quad y_o(k) \quad z_o(k) \,]$ at time $t_k = kT, k = 0, 1, ...n$ (T is the sampling interval), the corresponding mea-

[14]Following the estimation of the motion parameter vector, typically from a short observation interval, the IPP is carried out accounting for process noise, as in [34].

surements of azimuth and elevation are

$$\theta_m(k) = \theta(k,\Theta) + w_\theta(k) = \tan^{-1}\left(\frac{r_y(k,\Theta)}{r_x(k,\Theta)}\right) + w_\theta(k) \tag{3.2.10}$$

$$\phi_m(k) = \phi(k,\Theta) + w_\phi(k) \ = \tan^{-1}\left(\frac{r_z(k,\Theta)}{\sqrt{r_x^2(k,\Theta) + r_y^2(k,\Theta)}}\right) + w_\phi(k) \tag{3.2.11}$$

where w_θ and w_ϕ are independent zero-mean white Gaussian measurement noises with standard deviations (SD) σ_θ and σ_ϕ, respectively. The relative position coordinates of the projectile with respect to the passive sensor are

$$r_x(k,\Theta) \stackrel{\Delta}{=} x(k,\Theta) - x_o(k) \tag{3.2.12}$$

$$r_y(k,\Theta) \stackrel{\Delta}{=} y(k,\Theta) - y_o(k) \tag{3.2.13}$$

$$r_z(k,\Theta) \stackrel{\Delta}{=} z(k,\Theta) - z_o(k) \tag{3.2.14}$$

and

$$r(k,\Theta) \stackrel{\Delta}{=} \sqrt{r_x^2(k,\Theta) + r_y^2(k,\Theta) + r_z^2(k,\Theta)} \tag{3.2.15}$$

is the distance between the projectile and the passive sensor at time t_k.

In the problem considered, the azimuth and the elevation angles are known nonlinear functions of the target's position components $x(k,\Theta)$, $y(k,\Theta)$ and $z(k,\Theta)$ but the state of the projectile can only be evaluated numerically based on the following continuous-time nonlinear system (which assumes deterministic

motion with the unknown parameter vector Θ). This is

$$\dot{\mathbf{y}}(t,\Theta) = \mathbf{f}[\mathbf{y}(t,\Theta)] \tag{3.2.16}$$

where

$$\mathbf{f}[\mathbf{y}(t,\Theta)] = \begin{bmatrix} \dot{x}(t,\Theta) \\ \dot{y}(t,\Theta) \\ \dot{z}(t,\Theta) \\ \tau\cos\vartheta(t)\cos\varphi(t) + \alpha\alpha_m(t)D(t)\dot{x}(t,\Theta) \\ \tau\sin\vartheta(t)\cos\varphi(t) + \alpha\alpha_m(t)D(t)\dot{y}(t,\Theta) \\ \tau\sin\varphi(t) + \alpha\alpha_m(t)D(t)\dot{z}(t,\Theta) - g \end{bmatrix} \tag{3.2.17}$$

and in which $\alpha_m(t)$ is the Mach number-dependent drag coefficient multiplier, $D(t)$ is the drag coefficient factor and g is the standard acceleration due to gravity at sea level.

We have the LP full position obtained from intersecting the LoS measurement at time t_0 with the known-altitude plane $z = h$. This "flat-earth" model is the same as that in [34] except it does not have process noise.

The solution to the continuous-time nonlinear dynamic model described in equation (3.2.16) – (3.2.17) can be obtained recursively by the following 4th order Runge-Kutta method (RK) [20]

$$\mathbf{y}(t+\Delta t,\Theta) = \mathbf{y}(t,\Theta) + \frac{\Delta t}{6}(\xi_1 + 2\xi_2 + 2\xi_3 + \xi_4) \tag{3.2.18}$$

where Δt is a very small time interval and

$$\xi_1 = \mathbf{f}[\mathbf{y}(t, \Theta)] \tag{3.2.19}$$

$$\xi_2 = \mathbf{f}[\mathbf{y}(t + \frac{1}{2}\Delta t, \Theta) + \frac{\Delta t}{2}\xi_1] \tag{3.2.20}$$

$$\xi_3 = \mathbf{f}[\mathbf{y}(t + \frac{1}{2}\Delta t, \Theta) + \frac{\Delta t}{2}\xi_2] \tag{3.2.21}$$

$$\xi_4 = \mathbf{f}[\mathbf{y}(t + \Delta t, \Theta) + \Delta t\xi_3] \tag{3.2.22}$$

The result of the numerical integration of (3.2.16) yields the trajectory (including position and velocity information)

$$Y(\Theta) = \{\mathbf{y}(k, \Theta)\}_{k=0}^{n} \tag{3.2.23}$$

where, at the sampling time t_k, $\mathbf{y}(k, \Theta) \stackrel{\Delta}{=} \mathbf{y}(t_k, \Theta)$. We define, for later use, the stacked vector consisting of the position components $\mathbf{y}_p$ of $\mathbf{y}$ (the initial position was obtained based on the given LP altitude and the first LoS measurement)

$$G(\Theta) \stackrel{\Delta}{=} \begin{bmatrix} \mathbf{y}_p(1, \Theta) \\ \mathbf{y}_p(2, \Theta) \\ \cdots \\ \mathbf{y}_p(n, \Theta) \end{bmatrix} \tag{3.2.24}$$

where

$$\mathbf{y}_p(k, \Theta) \stackrel{\Delta}{=} [\, x(k, \Theta)\ y(k, \Theta)\ z(k, \Theta)\,]' \tag{3.2.25}$$

The available measurement vector over the trajectory (3.2.24) is denoted com-

pactly as (the first measurements $\theta_m(0)$ and $\phi_m(0)$ are used to estimate the LP location, subsequently assumed perfect)

$$Z^n \stackrel{\Delta}{=} \begin{bmatrix} \theta_m(1) \\ \phi_m(1) \\ \cdots \\ \theta_m(n) \\ \phi_m(n) \end{bmatrix} \stackrel{\Delta}{=} H\left[G(\Theta)\right] + \mathbf{w}_o \tag{3.2.26}$$

where the Gaussian noise vector $\mathbf{w}_o$ has the following $2n \times 2n$ covariance matrix

$$\mathbf{R} = \text{diag}[\ \sigma_\theta^2 \quad \sigma_\phi^2 \quad \cdots \quad \sigma_\theta^2 \quad \sigma_\phi^2\ ] \tag{3.2.27}$$

and G $(3n \times 1)$ is the trajectory (position only) from time t_1 to time t_n, given in (3.2.24).

Based on the noise-corrupted measurements Z^n, the goal is to find a target parameter vector estimate $\hat{\Theta}$ that uniquely determines the estimated 3-D trajectory.

3.3 Possible Ambiguity and the ML Estimator

As shown in Fig. 3.3.1, an ambiguity could exist due to the incomplete observability: when two or more different 3-D trajectories are projected into the 2-D sensor focal plane, they may "look" the same to the passive sensor. This could yield two or more different target parameter vectors corresponding to the same trajectory.

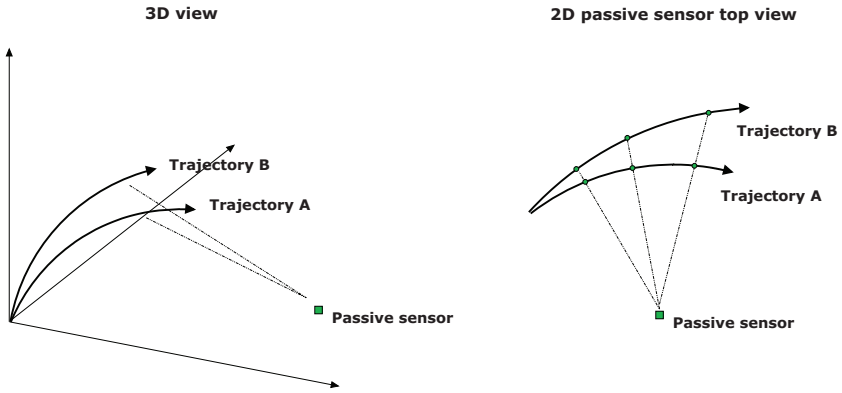


Figure 3.3.1: Possible ambiguity of 3-D trajectory using 2-D LoS measurements (both top view and side view, the latter not shown, have the same LoS for all points on the trajectory).

The LF of the target parameter vector Θ given the total history of the passive sensor measurements Z^n, which depends on the parameter Θ through the trajectory (position only) $G(\Theta)$, is

$$\Lambda(\Theta) = p\,(Z^n|\Theta) = p\,(Z^n|G(\Theta)) = \prod_{k=1}^{n} p\left[\theta_m(k), \phi_m(k)|\mathbf{y}_p(k,\Theta)\right] \qquad (3.3.1)$$

where, in view of (3.2.10) and (3.2.11),

$$p\left(\theta_m(k), \phi_m(k)|\mathbf{y}_p(k,\Theta)\right) = \mathcal{N}\left[\theta_m(k); \theta(k,\Theta), \sigma_\theta^2\right]\mathcal{N}\left[\phi_m(k); \phi(k,\Theta), \sigma_\phi^2\right] \qquad (3.3.2)$$

The ML estimate of Θ, in view of (3.3.1), follows from the following nonlinear least squares (NLS) problem

$$\hat{\Theta} = \arg\max_{\Theta} \Lambda(\Theta) = \arg\min_{\Theta} \lambda(\Theta) \qquad (3.3.3)$$

where

$$\lambda(\Theta) \triangleq \sum_{k=1}^{n} \left\{ \frac{1}{2\sigma_\theta^2} \left[\theta_m(k) - \theta(k,\Theta)\right]^2 + \frac{1}{2\sigma_\phi^2} \left[\phi_m(k) - \phi(k,\Theta)\right]^2 \right\} \tag{3.3.4}$$

is the negative log-likelihood function (NLLF) with the irrelevant additive constants omitted.

For any specific Θ, we can obtain

1) the trajectory (position and velocity) $Y(\Theta)$ (3.2.23) by the RK algorithm,
2) the trajectory (position only) $G(\Theta)$ (3.2.24),
3) the azimuth angles $\theta(k,\Theta)$ and elevation angles $\phi(k,\Theta)$ in (3.2.10) and (3.2.11),

and, with the noisy measurements,

4) the NLLF (3.3.4).

The ML estimate can be found by seeking the minimum of the NLLF over the target parameter space. This can be done by using a global optimization method (say, a genetic algorithm) or, preferably in the present venue, with a search strategy consisting of two stages, as described in Section 3.4. The necessary derivatives in the estimability analysis and numerical search are (i) the derivatives of the function H with respect to G and (ii) the derivatives of the function G with respect to Θ. While (i) can be obtained analytically, (ii) can be obtained only numerically. This numerical difficulty necessitates (high-fidelity) approximations of the true FIM and CRLB and also induces additional

challenges to the numerical search.

3.4 FIM, CRLB and Numerical Search with Two Stages

Estimability is determined by checking the invertibility of FIM of Θ from the LF (3.3.1). Assuming white Gaussian measurement noise, we have the FIM (to be evaluated at the true Θ [2])

$$\begin{aligned}\mathbf{F}_{\Theta}(n) &= E\left\{\nabla_{\Theta}\ln p\left[Z^{n}|G(\Theta)\right]\left(\nabla_{\Theta}\ln p\left[Z^{n}|G(\Theta)\right]\right)'\right\}\\ &= E\left\{\nabla_{\Theta}\ln p_{\mathbf{w}}\left[Z^{n}-H\left[G(\Theta)\right]|\Theta\right]\left(\nabla_{\Theta}\ln p_{\mathbf{w}}\left[Z^{n}-H\left[G(\Theta)\right]|\Theta\right]\right)'\right\}\\ &= \nabla_{\Theta}H\left[G(\Theta)\right]\mathbf{R}^{-1}\left(\nabla_{\Theta}H\left[G(\Theta)\right]\right)'\\ &= \nabla_{\Theta}G(\Theta)\,\nabla_{G}H(G)\,\mathbf{R}^{-1}\left(\nabla_{G}H(G)\right)'\left(\nabla_{\Theta}G(\Theta)\right)'\end{aligned} \tag{3.4.1}$$

where n is the number of measurements in (3.2.26) (their statistics and geometry are not indicated for brevity). The gradient $\nabla_{G}H(G)$ can be obtained analytically, given the trajectory (position only) $G(\Theta)$, while the elements of $\nabla_{\Theta}G(\Theta)$ can be obtained (only) numerically as follows

$$\left[\nabla_{\Theta}G\right]_{ij}\approx\frac{G_{j}(\Theta_{i}+\Delta\Theta_{i})-G_{j}(\Theta_{i})}{\Delta\Theta_{i}}\qquad i=1,...,4;\;\;j=1,...,3n \tag{3.4.2}$$

where the multiplier Δ is a very small value, G_j is the jth element of G (consisting of 3 position components over n sampling times) and Θ_i is the ith element of Θ (the 4-D target parameter vector).

From (3.2.10), (3.2.11), (3.2.24), (3.2.25) and (3.2.26), we have the following

expressions that enter into the FIM

$$H_x^{\theta}\left[G(k,\Theta)\right] \triangleq \frac{\partial\theta}{\partial x}=\frac{-r_y(k,\Theta)}{r_x^2(k,\Theta)+r_y^2(k,\Theta)} \tag{3.4.3}$$

$$H_y^{\theta}\left[G(k,\Theta)\right] \triangleq \frac{\partial\theta}{\partial y}=\frac{r_x(k,\Theta)}{r_x^2(k,\Theta)+r_y^2(k,\Theta)} \tag{3.4.4}$$

$$H_z^{\theta}\left[G(k,\Theta)\right] \triangleq \frac{\partial\theta}{\partial z}=0 \tag{3.4.5}$$

$$H_x^{\phi}\left[G(k,\Theta)\right] \triangleq \frac{\partial\phi}{\partial x}=\frac{-r_x(k,\Theta)r_z(k,\Theta)}{r^2(k,\Theta)\sqrt{r_x^2(k,\Theta)+r_y^2(k,\Theta)}} \tag{3.4.6}$$

$$H_y^{\phi}\left[G(k,\Theta)\right] \triangleq \frac{\partial\phi}{\partial y}=\frac{-r_y(k,\Theta)r_z(k,\Theta)}{r^2(k,\Theta)\sqrt{r_x^2(k,\Theta)+r_y^2(k,\Theta)}} \tag{3.4.7}$$

$$H_z^{\phi}\left[G(k,\Theta)\right] \triangleq \frac{\partial\phi}{\partial z}=\frac{\sqrt{r_x^2(k,\Theta)+r_y^2(k,\Theta)}}{r^2(k,\Theta)} \tag{3.4.8}$$

Once the FIM is found to be invertible (numerically), the target parameter vector Θ is estimable. We can obtain the corresponding CRLB (for the n measurements), which provides (if the estimator is efficient) the accuracy of the target parameters' estimate

$$\mathbf{C}_{\Theta}(n) \triangleq \mathbf{F}_{\Theta}^{-1}(n) \tag{3.4.9}$$

and we define the corresponding CRLB-based standard deviation (SD) for the

estimates of the target parameters as

$$\begin{aligned}
\sigma_{\vartheta_0}(n) &\triangleq \sqrt{\mathbf{C}_{\Theta,\vartheta_0}(n)} \\
\sigma_{\varphi_0}(n) &\triangleq \sqrt{\mathbf{C}_{\Theta,\varphi_0}(n)} \\
\sigma_{\alpha}(n) &\triangleq \sqrt{\mathbf{C}_{\Theta,\alpha}(n)} \\
\sigma_{\tau}(n) &\triangleq \sqrt{\mathbf{C}_{\Theta,\tau}(n)}
\end{aligned} \tag{3.4.10}$$

where the lower-case subscripts above indicate the diagonal elements of (3.4.9).

Under the unknown LP altitude assumption, the associated FIM and CRLB for the augmented target parameter vector are accordingly extended in the similar manner. We have the FIM (to be evaluated at the true $\bar{\Theta}$)

$$\bar{\mathbf{F}}_{\bar{\Theta}}(n) = \nabla_{\bar{\Theta}}\bar{G}(\bar{\Theta})\,\nabla_{\bar{G}}H(\bar{G})\,\mathbf{R}^{-1}\,\left(\nabla_{\bar{G}}\,H(\bar{G})\right)'\,\left(\nabla_{\bar{\Theta}}\,\bar{G}(\bar{\Theta})\right)' \tag{3.4.11}$$

where

$$[\nabla_{\bar{\Theta}}\bar{G}]_{ij} \approx \frac{\bar{G}_j(\bar{\Theta}_i + \Delta\bar{\Theta}_i) - \bar{G}_j(\bar{\Theta}_i)}{\Delta\bar{\Theta}_i} \quad i = 1,...,5;\;\; j = 1,...,3n \tag{3.4.12}$$

with $\bar{G}_j$ the jth element of $\bar{G}$ and $\bar{\Theta}_i$ is the ith element of $\bar{\Theta}$ (the 5-D augmented target parameter vector).

Once the FIM is found to be invertible, we can obtain an estimable augmented target parameter vector and the corresponding CRLB is

$$\bar{\mathbf{C}}_{\bar{\Theta}}(n) \triangleq \bar{\mathbf{F}}_{\bar{\Theta}}^{-1}(n) \tag{3.4.13}$$

The CRLB-based SDs for the estimate of the components of the augmented target parameter vector are defined as

$$\begin{aligned}
\bar{\sigma}_{\vartheta_0}(n) &\triangleq \sqrt{\bar{\mathbf{C}}_{\bar{\Theta},\vartheta_0}(n)} \\
\bar{\sigma}_{\varphi_0}(n) &\triangleq \sqrt{\bar{\mathbf{C}}_{\bar{\Theta},\varphi_0}(n)} \\
\bar{\sigma}_{\alpha}(n) &\triangleq \sqrt{\bar{\mathbf{C}}_{\bar{\Theta},\alpha}(n)} \\
\bar{\sigma}_{\tau}(n) &\triangleq \sqrt{\bar{\mathbf{C}}_{\bar{\Theta},\tau}(n)} \\
\bar{\sigma}_{h}(n) &\triangleq \sqrt{\bar{\mathbf{C}}_{\bar{\Theta},h}(n)}
\end{aligned} \tag{3.4.14}$$

where the lower-case subscripts above indicate the diagonal elements of (3.4.13). Even with knowledge of the LP altitude, concavity of the LF is not guaranteed and a (single stage) direct continuous search over the whole target parameter space could converge to a local maximum [29][30]. We adopt a search strategy with two stages described in the sequel: (i) mixed search stage — with a grid in the drag-thrust/drag-thrust-altitude (DT/DTA) subspace and continuous in the launch angles subspace, followed by (ii) continuous search in the whole target parameter (DT/DTA+launch angles) space.

We first discuss the case with known LP altitude.

- In the first – mixed-search – stage, with prior knowledge that both drag coefficient and thrust are in a specified intervals, we work according to a 2-D drag-thrust grid (DTG). For each grid point, a continuous numerical search[15] over the launch angles space (the azimuth and elevation 2-D

[15]The MATLAB *fmincon* active-set algorithm was used in the following simulation study.

subspace of Θ, $\Phi \stackrel{\Delta}{=} [\,\vartheta_0 \quad \varphi_0\,]'$) is carried out. Following this, the associated LFs at each grid point are compared. Choosing the launch angles' estimate corresponding to the DTG point with the highest likelihood, we get, so far, the best estimate of the target parameter vector Θ, denoted as $\hat{\Theta}^m$. If necessary, the DTG can be re-centered and refined, and the procedure repeated until some termination criterion be met.

- In the second stage, a continuous search over the whole 4-D target parameter space is carried out within a smaller re-centered region. The contracted search region is centered at $\hat{\Theta}^m$ and sized with boundaries of plus and minus one step size (from the previous mixed search) in each coordinate of the drag-thrust subspace and specified $\Delta\vartheta$, $\Delta\varphi$ in the launch angles subspace.

With unknown LP altitude, a similar numerical search with two stages is carried out but with a drag-thrust-altitude grid (DTAG) in the first stage and a 5-D space of the augmented target parameter vector to be dealt with in the second stage.

3.5 Simulations

The true trajectory is generated based on the deterministic motion model described in (3.2.16) – (3.2.17). The target parameters are set as follows: the true launch azimuth angle $\vartheta_0 = 45°$, the true launch elevation angle $\varphi_0 = 45°$, the true drag coefficient $\alpha = 0.03\ \mathrm{m^2/kg}$, the true thrust $\tau = 200\,\mathrm{m/s^2}$ and the true LP altitude $h = 200\,\mathrm{m}$. The BoT is at 4 s after the launch. The azimuth

and elevation measurements have the same SD over the observation interval, $\sigma_\theta = \sigma_\phi = 1$ mrad. The sampling interval is $T = 0.1$ s.

For the resulting trajectory: the LP location of the projectile was at [500 500 200] m, the projectile range to impact (at sea level) is 26887 m, the impact time is at 69.7 s (about 66 s in the ballistic mode), the maximum speed is 742 m/s, the speed at impact is 359 m/s and the apogee is 5770 m.

As the sample scenario shown in Fig. 3.5.1, we consider three different platform movements: MP1 with constant velocity [35 35 0] m/s (i.e., NE 45° with speed 50 m/s), MP2 with constant velocity [−35 −35 0] m/s (i.e., SW 45° with speed 50 m/s) and MP3 with constant velocity [70 70 0] m/s (i.e., NE 45° with speed 100 m/s), respectively. All these platforms start (for the MPs) or are fixed (for the SP) in the close scenario at [5000 0 2000] m or in the long distance scenario at [10000 0 2000] m.

3.5.1 CRLB Evaluation Results

We first examine, with the aid of the CRLB (and its corresponding SD), how knowledge of the LP altitude affects the target parameter estimation performance. In the long distance scenario with the stationary passive sensor fixed at [10000 0 2000] m, the FIMs of the target parameter vector Θ and the augmented target parameter vector $\bar{\Theta}$ are both invertible. Fig. 3.5.2 shows the CRLB-based SD as function of the observation time (OT) (from 6 s to 10 s, which gives 2 – 6 s in the ballistic mode) for the azimuth, elevation, drag coef-

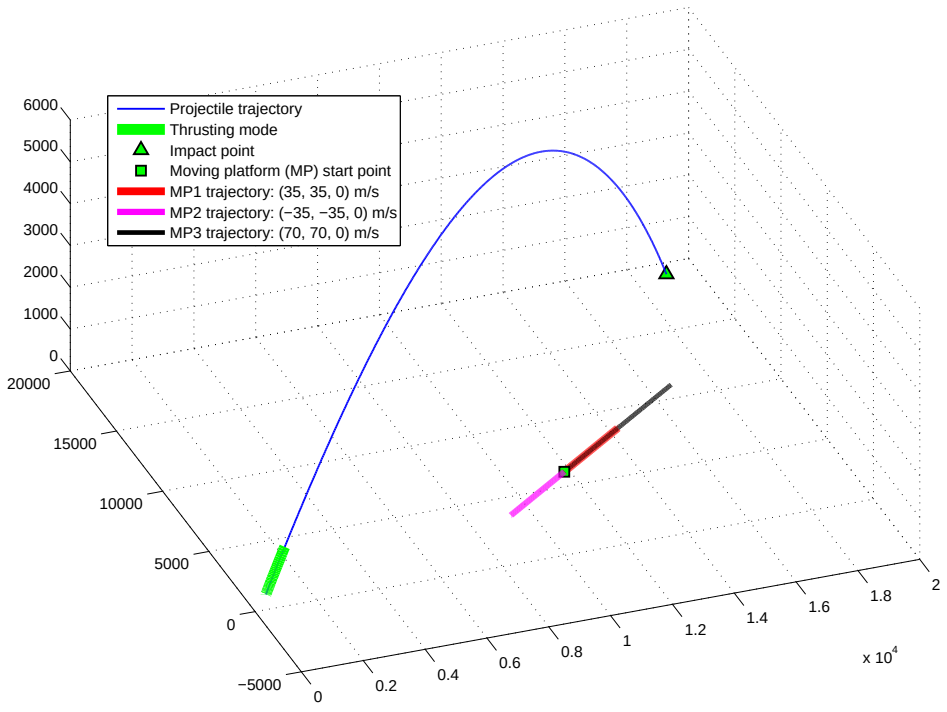


Figure 3.5.1: Sample scenario with various platform behaviors.

ficient and thrust, respectively. It can be seen that if the LP altitude is known, the CRLB is reduced. Fig. 3.5.3 shows the CRLB-based SD of the additional parameter, the unknown LP altitude, as function of the OT (with the truth $h = 200\,\text{m}$).

We analyze how the projectile-sensor geometry affects estimability with a stationary passive sensor fixed at different locations: in the close scenario at $[\,5000 \quad 0 \quad 2000\,]\,\text{m}$ and in the long distance scenario at $[\,10000 \quad 0 \quad 2000\,]\,\text{m}$, respectively. With the LP altitude known as $h = 200\,\text{m}$, the FIMs of the target parameter vector Θ given the OT as $6-10\,\text{s}$ for both the close and long distance scenarios are invertible. Fig. 3.5.4 shows the resulting CRLB-based SD

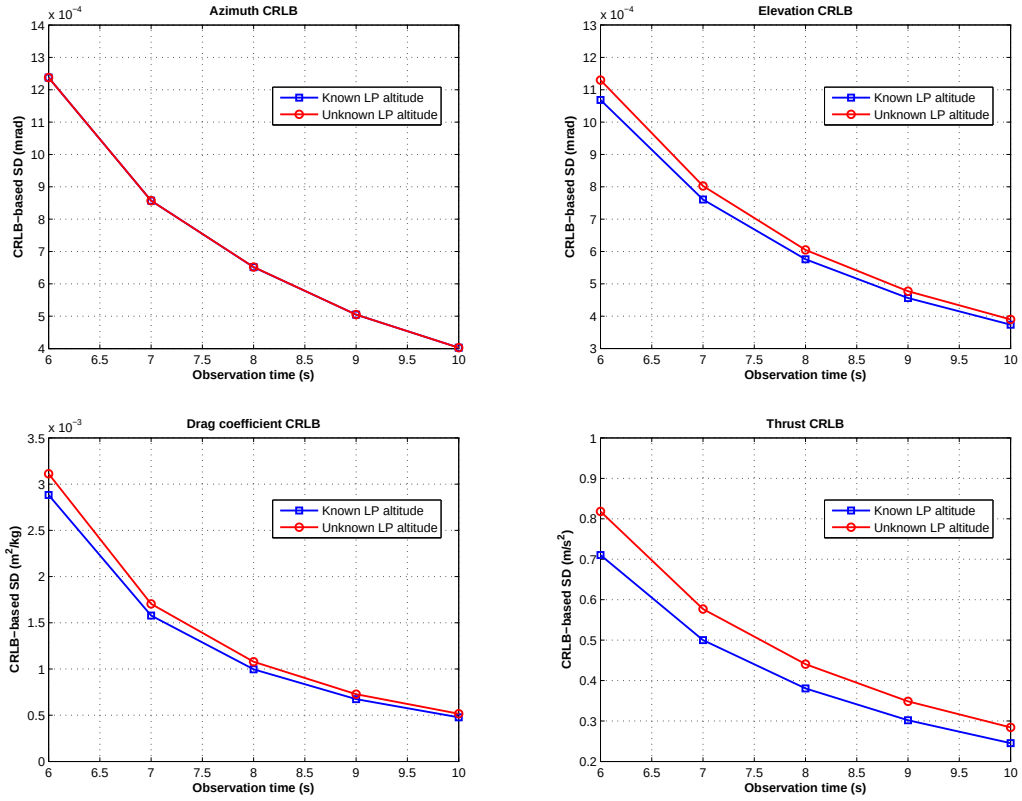


Figure 3.5.2: CRLB-based SD as function of OT for target parameters with known LP altitude vs. unknown LP altitude ($h = 200$ m)

as function of the OT for the azimuth, elevation, drag coefficient and thrust, respectively. It can be seen that the CRLB is approximately proportional to the projectile-sensor distance: doubling the projectile-sensor distance approximately doubles the corresponding CRLB-based SD. Given the OT = 8 s (which is about 11% of the flight time) one may find an efficient estimator that achieves highly accurate estimates for each component of the target parameter vector. The unknown LP altitude case yields the similar difference in the close and long distance scenarios for each component of $\bar{\Theta}$.

We examine how the platform behavior affects the estimability for three dif-

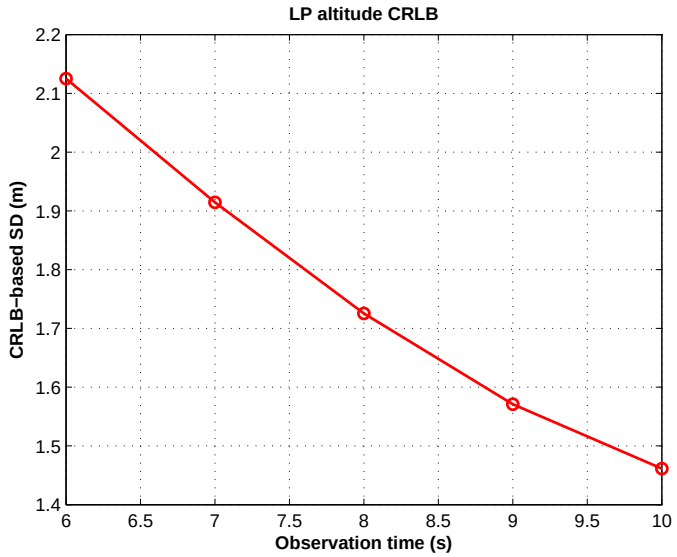


Figure 3.5.3: CRLB-based SD as function of OT for the unknown LP altitude (truth $h = 200\,\text{m}$)

ferent moving scenarios (MP1, MP2 and MP3) and one SP. Fig. 3.5.5 shows, in the long-distance scenario without knowing the LP altitude (but the truth is $h = 200\,\text{m}$), the CRLB-based SD of different target parameters as the function of OT for the MPs and the SP. It can be seen that there is possible, though not major, improvement of MP over the SP. Similar improvement of the MPs over the SP exists when the LP altitude is known.

3.5.2 Results of Numerical Search with Two Stages

In the following long-distance scenario with the stationary passive sensor fixed at $[\,10000 \quad 0 \quad 2000\,]\,\text{m}$, the numerical search, implemented by the MATLAB embedded *fmincon* active-set algorithm, is carried out with two stages. In the first (mixed search) stage, we use the coarse grid (with the search range and stepsize settings for each target parameter) shown in Table 3.5.14. For each grid

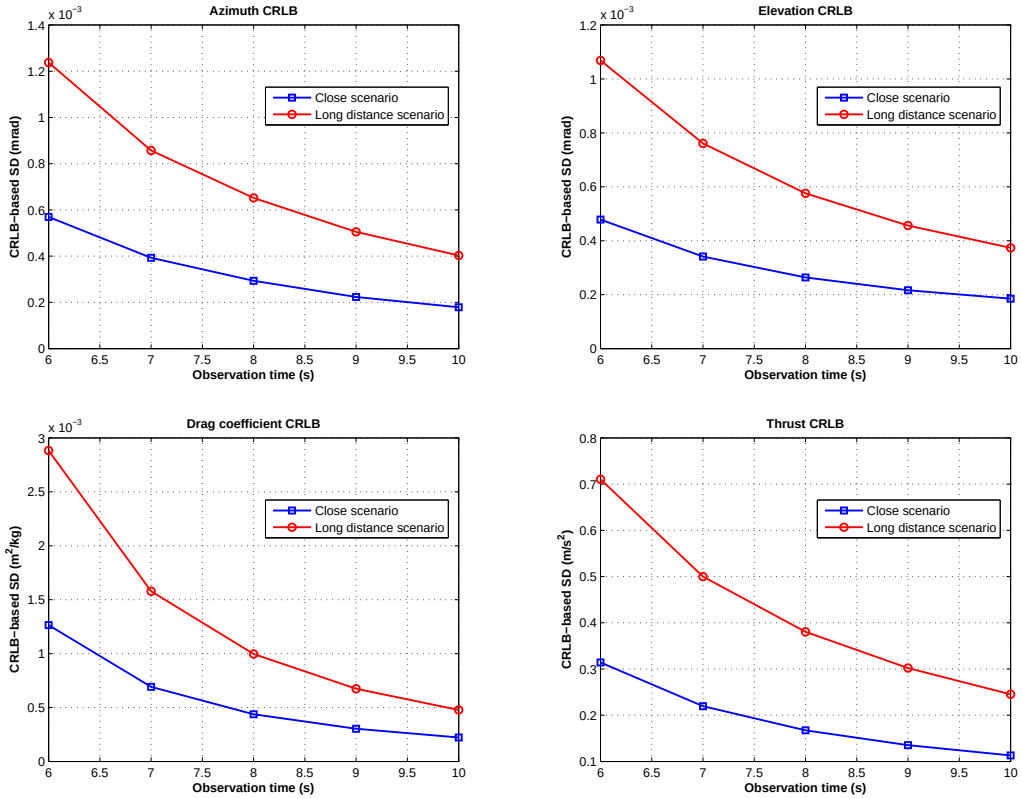


Figure 3.5.4: CRLB-based SD as function of OT for target parameter vector (with known LP altitude as 200 m) in both the close- and the long-distance scenarios: stationary sensor located at [5000 0 2000] m and at [10000 0 2000] m, respectively.

point, the numerical search is initialized at $\hat{\Phi}_0 = [\, 40^\circ \quad 40^\circ \,]'$ and terminated by the specified criteria for the tolerance on the target parameters[16] (norm of the change of Θ between consecutive iterations) and the tolerance on the objective function[17] (norm of the change of NLLF between consecutive iterations): under the known LP altitude assumption, as 10^{-6} and 10^{-6}, respectively; under the unknown LP altitude assumption, as 10^{-10} and 10^{-10}, respectively. In the second

[16]The tolerance is set by the option 'TolX' in MATLAB; to get more precision on the parameters, lower 'TolX' is used.

[17]The tolerance is set by the option 'TolFun' in MATLAB; to find a smaller minimum, lower 'TolFun' is used.

Table 3.5.14: Search range and grid stepsize settings (for the 6×6 DTG and the $6 \times 6 \times 3$ DTAG)

True value	$\vartheta = 45(^\circ)$	$\varphi = 45(^\circ)$	$\alpha = 0.03\,(\mathrm{m}^2/\mathrm{kg})$	$\tau = 200\,(\mathrm{m/s}^2)$	$h = 200\,(\mathrm{m})$
Search range	$[\,\vartheta_{\min} \quad \vartheta_{\max}\,]$	$[\,\varphi_{\min} \quad \varphi_{\max}\,]$	$[\,\alpha_{\min} \quad \alpha_{\max}\,]$	$[\,\tau_{\min} \quad \tau_{\max}\,]$	$[\,h_{\min} \quad h_{\max}\,]$(if ukn.)
	$[\,20 \quad 160\,]$	$[\,25 \quad 80\,]$	$[\,0.02 \quad 0.04\,]$	$[\,150 \quad 250\,]$	$[\,50 \quad 300\,]$
Stepsize	N/A	N/A	0.004	20	125

stage, the continuous search is initialized at $\hat{\Theta}^m$ (or $\hat{\tilde{\Theta}}^m$, if the LP altitude is unknown) and terminated, under both the known and unknown LP altitude assumptions, by the criteria for the tolerance on the target parameters as 10^{-12} and the tolerance on the objective function NLLF as 10^{-12}. The contracted search region is recentered at $\hat{\Theta}^m/\hat{\tilde{\Theta}}^m$ and resized with boundaries of plus and minus one stepsize (from the mixed search) in each coordinate of the DT/DTA subspace and $\Delta\vartheta = \Delta\varphi = 10^\circ$ in the launch angles subspace.

Fig. 3.5.6 illustrates the LF by plotting the inverse NLLF of the launch angles for selected DTG points (with known LP altitude) and selected DTAG points (with unknown LP altitude), respectively. It can be seen that, whether the LP altitude is known or not, the surface of the LF is not only nonconcave but also can be bimodal. As one gets closer to the truth the inverse NLLF surface of the launch angles becomes unimodal. This allows the mixed search to converge to the unique global maximum of the LF over the grid points with high likelihood (i.e., low NLLF).

A single-run study is conducted first. Table 3.5.15 shows, with the known LP altitude (as 200 m), the numerical search results with the OT of 8 s and 10 s

(about 11% and 14% of the flight time), respectively. For each given OT, the best result obtained from the first (mixed search) stage is the one corresponding to the highest likelihood (the lowest NLLF). As shown in the bottom part of Table 3.5.15 (cf. the truth Θ = [45 45 0.03 200]'), the search strategy with two stages finally converges at $\hat{\Theta}$ = [44.3 44.5 0.036 204]' for OT = 8 s and at $\hat{\Theta}$ = [42.8 44.2 0.036 205]' for OT = 10 s, respectively. These lead to the final IPP errors as 1981 m and 2065 m, i.e., the IPP percentages (the ratio between the IPP error and the projectile range) as about 7.4% and 7.7%, respectively. As shown in Table 3.5.16 (cf. the truth $\bar{\Theta}$ = [45 45 0.03 200 200]'), under the unknown LP altitude assumption, the numerical search with two stages finally converges at $\hat{\bar{\Theta}}$ = [45.1 45.6 0.036 191 298]' for OT = 8 s and $\hat{\bar{\Theta}}$ = [45.4 46.0 0.0292 187 300]' for OT = 10 s. These yield the IPP errors as 3320 m and 1320 m, which are about 12.3% and 4.9% of the projectile range, respectively.

A Monte Carlo study (with 100 runs) is conducted for both the known LP altitude and unknown LP altitude assumptions. The estimation results (the root mean squared error, RMSE, for each parameter) and IPP results (the IPP error and the IPP percentage) with different OTs are shown in Table 3.5.17. It can be seen that, under the same assumption about the LP altitude, increasing the available observation time improves the estimation and IPP performance — observation time is a key factor [34]; given the same amount of observation time, compared to the known LP altitude case, the lack of knowledge of the LP

Table 3.5.15: Results of numerical search with two stages (single run, known LP altitude, the 4-D target parameter vector truth is Θ = **[45 45 0.03 200]'**)

OT = 8 s					OT = 10 s				
First (mixed search) stage									
DTG ($\hat{\alpha}$, $\hat{\tau}$)	$\hat{\vartheta}_0$	$\hat{\varphi}_0$	NLLF	Iter.	DTG ($\hat{\alpha}$, $\hat{\tau}$)	$\hat{\vartheta}_0$	$\hat{\varphi}_0$	NLLF	Iter.
(0.040, 210)	***41.6***	***43.6***	**227**	8	*(0.040, 210)*	***40.8***	***43.5***	**225**	9
(0.036, 210)	40.6	43.4	374	9	(0.036, 210)	39.3	43.2	996	9
(0.032, 210)	39.7	43.2	650	9	(0.020, 190)	51.8	46.6	1160	9
(0.028, 190)	54.7	46.9	816	16	(0.032, 210)	37.9	42.9	2770	9
(0.028, 210)	38.8	43.0	1060	9	(0.024, 190)	53.6	46.8	3500	10
Second stage									
$\hat{\Theta}$ = ***[44.3 44.5 0.036 204]'***					$\hat{\Theta}$ = ***[42.8 44.2 0.036 205]'***				
IPP error = 1981 m					IPP error = 2065 m				
IPP percentage= 7.4%					IPP percentage = 7.7%				

altitude worsens the corresponding IPP results. However, for the coarse grid, whether the LP altitude is known or not, once given a good enough observation time (≥8 s, 11% of the flight time), we can achieve IPP of good accuracy (with the IPP error around 10% of the projectile range).

Table 3.5.17 also summarizes the computational aspect[18] of the numerical approach with two stages. The (approximate) computation time of the mixed search (in the first stage) at each grid point (of the 6×6 DTG or the $6 \times 6 \times 3$ DTAG) is T^m and of the continuous search (in the second stage) is T^d. It can be seen, if the mixed search is implemented to be parallelizable, the overall computation time for these various situations (different OTs and/or different assumptions about the LP altitude) would be only slightly above one minute

[18]The computation time data are collected based on the MATLAB-embedded *fmincon* code running on an Intel i5 PC.

Table 3.5.16: Results of numerical search with two stages (single run, unknown LP altitude, the 5-D target parameter vector truth is $\bar{\Theta}$ = **[45 45 0.03 200 200]'**)

OT = 8 s						OT = 10 s					
First (mixed search) stage											
DTAG $\left(\hat{\alpha},\ \hat{\tau},\ \hat{h}\right)$	$\hat{\vartheta}_0$	$\hat{\varphi}_0$	NLLF	Iter.		DTAG $\left(\hat{\alpha},\ \hat{\tau},\ \hat{h}\right)$	$\hat{\vartheta}_0$	$\hat{\varphi}_0$	NLLF	Iter.	
(0.040, 210, 175)	***43.1***	***44.0***	**179**	10		*(0.032, 190, 300)*	***43.9***	***45.6***	**224**	12	
(0.032, 190, 300)	44.9	45.6	205	11		(0.040, 210, 175)	42.4	44.0	232	11	
(0.040, 230, 50)	40.3	42.3	242	8		(0.020, 210, 50)	46.4	44.7	336	12	
(0.036, 210, 175)	42.1	43.8	247	11		(0.036, 190, 300)	45.6	45.9	463	13	
(0.040, 190, 300)	47.0	46.0	304	13		(0.036, 210, 175)	40.9	43.7	482	10	
Second stage											
$\hat{\bar{\Theta}}$ = *[* ***45.1***	***45.6***	***0.036***	***191***	***298]'***		$\hat{\bar{\Theta}}$ = *[* ***45.4***	***46.0***	***0.0292***	***187***	***300]'***	
IPP error = 3320 m						IPP error = 1320 m					
IPP percentage = 12.3%						IPP percentage = 4.9%					

Table 3.5.17: Results of numerical search with two stages (100 MC runs)

LP	Obsv.	RMSE					IPP err. (m)	IPP pct.	T^m (s)	T^d (s)
Alt.	time	ϑ_0 (°)	φ_0 (°)	α (m^2/kg)	τ (m/s^2)	h (m)				
Kn.	8 s	2.73	0.94	0.0081	6.81	—	2722	10.1%	15	47
	10 s	2.32	0.86	0.0070	6.13	—	2400	8.9%	17	50
Ukn.	8 s	3.90	0.60	0.0077	9.18	86	3549	13.2%	16	50
	10 s	2.76	1.33	0.0068	9.69	82	3024	11.2%	20	52

(with MATLAB, but it can be decreased by one or two orders of magnitude with C/C++ or with direct implementation hardware). The search strategy with two stages is real-time implementable.

3.6 Summary and Conclusions

Estimability of a 3-D trajectory using 2-D measurements from a single passive

sensor (stationary or moving with constant velocity) is investigated. Two major assumptions about the important LP altitude information were considered: LP altitude known or unknown. The FIM and the CRLB of the target parameter vector that uniquely determines its trajectory were obtained. With/without knowledge of the LP altitude, the full-rank FIM and the corresponding CRLB show that we can estimate the trajectory in 3-D from a single passive sensor for the final IPP effectively using a short observation interval (10% – 15% of the flight time is good enough for many situations). A moving platform could have certain, though not major, benefits over a stationary platform. Facing a challenging nonlinear problem with unavoidable numerical integration and derivative evaluation, a numerical search strategy with two stages was proposed. The mixed search appears to offer excellent results, and since it is efficiently parallelizable we propose that we have a real-time implementable solution.

The methodology used for IPP with known LP altitude, using the search strategy with two stages is summarized in Fig. 3.6.1. The unknown LP altitude case is similar but using DTAG.

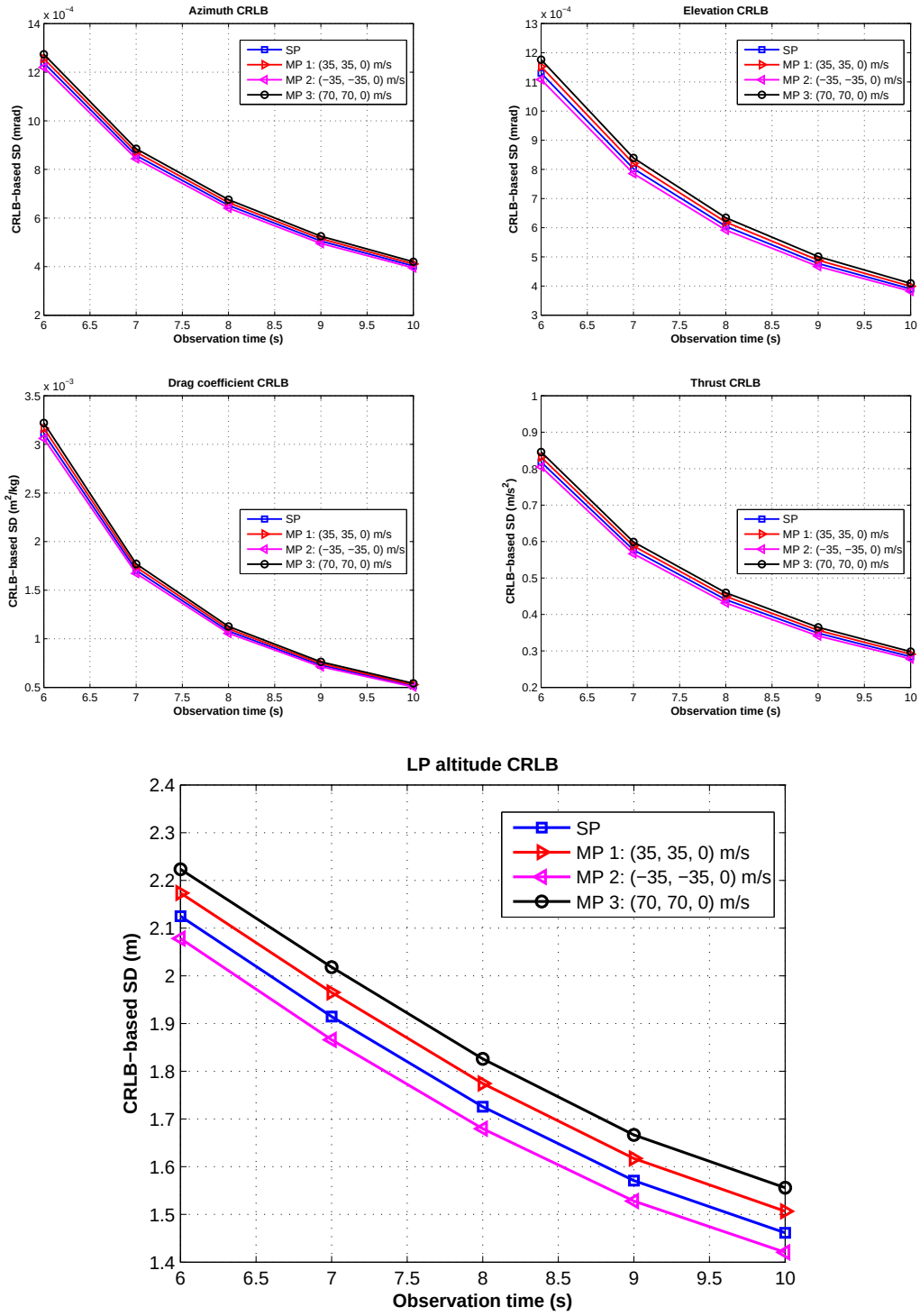
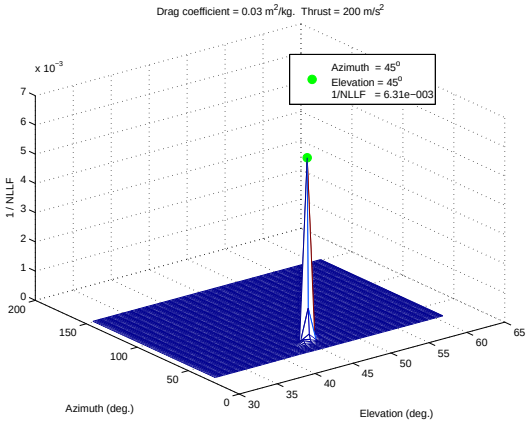
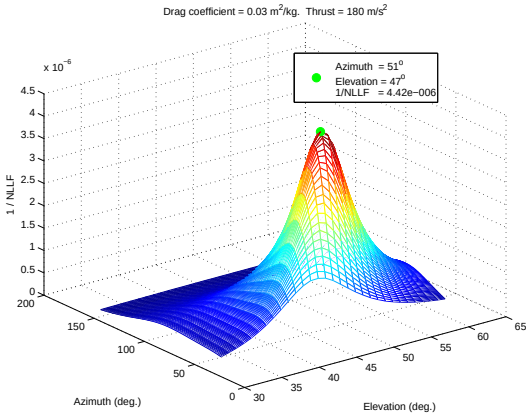


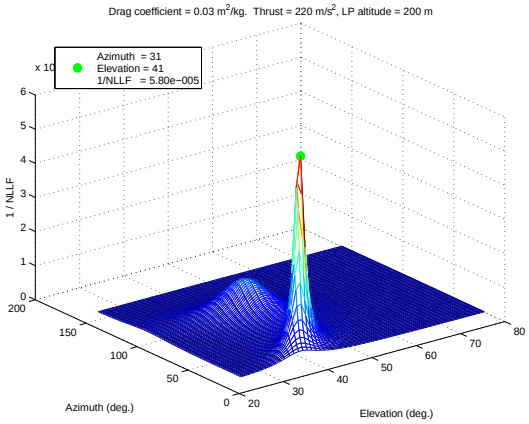
Figure 3.5.5: CRLB-based SD as function of OT for augmented target parameter vector (with unknown LP altitude) for the MPs and the corresponding SP.



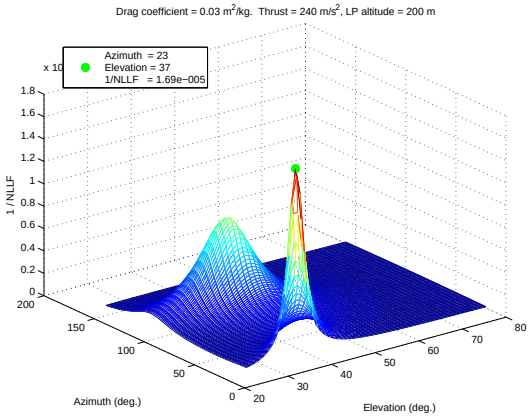
(a) DTG point (0.03 200)



(b) DTG point (0.03 180)



(c) DTAG point (0.03 220 200)



(d) DTAG point (0.03 240 200)

Figure 3.5.6: Surface of inverse NLLF vs. launch angles for selected DTG/DTAG points

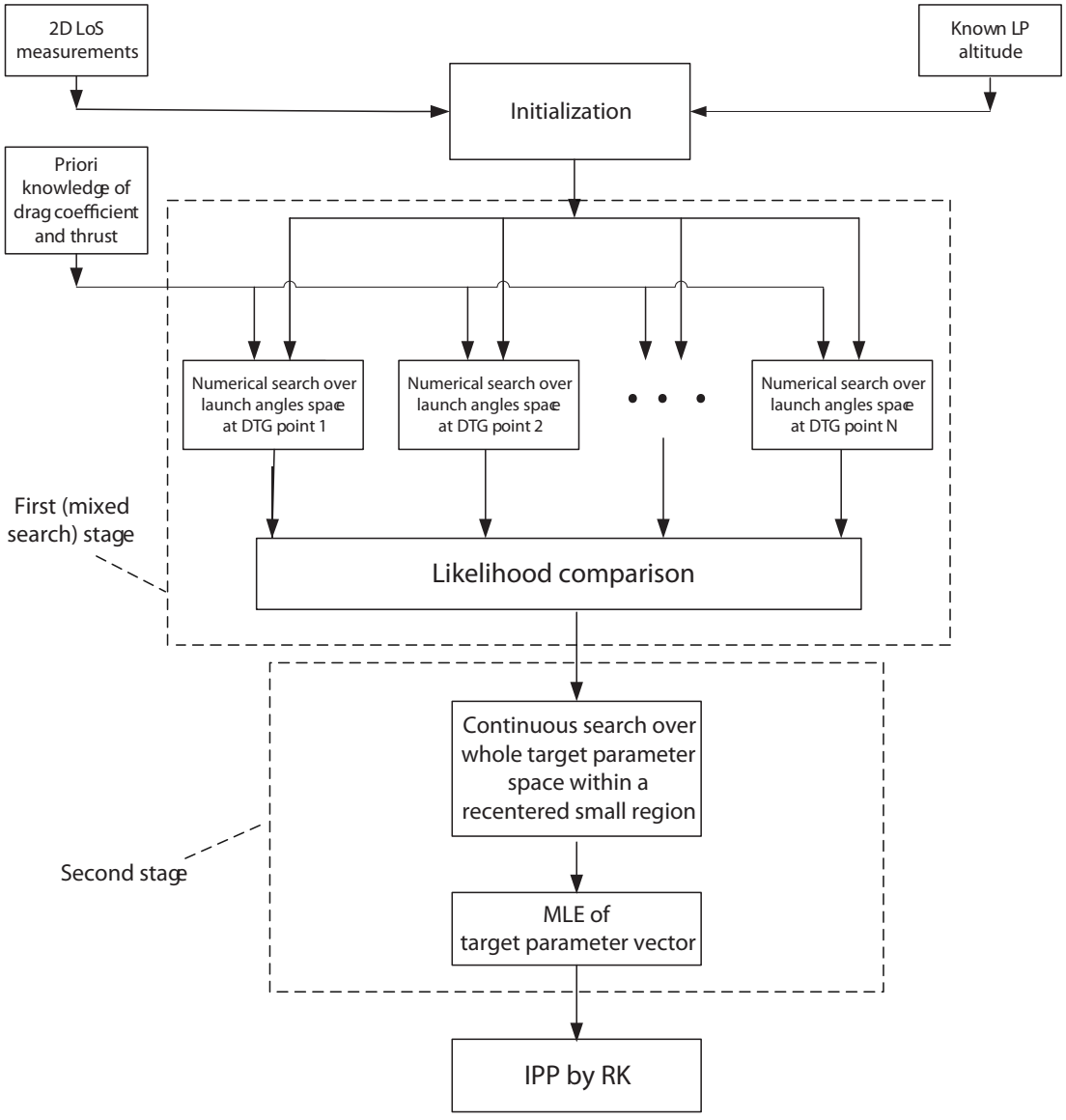


Figure 3.6.1: IPP procedure using numerical search with two stages (with known LP altitude)

Chapter 4

Heterogeneous Track-to-Track Fusion

Track-to-track fusion using estimates from multiple sensors can achieve better estimation performance than single sensor tracking. If the local sensors use different system models in different state spaces, the problem of heterogeneous track-to-track fusion arises. Compared with homogeneous track-to-track fusion that assumes the same system model for different sensors, the heterogeneous case poses two major challenges. The first one is that we have to fuse estimates from different state spaces (related by a certain nonlinear transformation). The second is the estimation errors' dependence problem due to the "common process noise effect" and there is no known way to capture the "common" part exactly. Two heterogeneous track-to-track fusion approaches, namely, the linear minimum mean square error approach and the maximum likelihood approach, are presented and compared with the corresponding centralized measurement tracker/fuser (also known as measurement-to-track fuser).

4.1 Introduction

In a multisensor tracking system the best target state estimation performance

is obtained by a centralized tracker/fuser (CTF), by directly sending to the fusion center (FC) all the measurements of the local sensors.[19] However, in many practical situations, because of communication constraints, each local sensor has its own information processing system and sends only tracks to the FC, which fuses appropriately tracks from different local sensors to achieve comparable estimation performance to that of the CTF [3].

For track-to-track fusion (T2TF) from homogeneous local trackers (which use the same target state space), the "common process noise effect" (quantified by the crosscovariance matrix) has been theoretically well-established [3]. However, there is no known way for the calculation of the crosscovariance matrix in the case of heterogeneous local trackers (which use the different target state spaces). The difficulty to evaluate the crosscovariance matrix in the heterogeneous case is that it requires to capture the "common" part of process noises from different state spaces to quantify the crosscorrelation.

In the literature there are few works dealing with the model heterogeneity. A heterogeneous T2TF fusion approach was presented in [6] to fuse the tracks from an active sensor and a passive sensor with different state vectors. However, the fusion was done by using the full Cartesian state estimates (from an active sensor) to update the smaller angular state estimates (from a passive sensor). An expression for the steady state crosscovariance matrix for dissimilar sensors (of the same state vector but with different measurement noise

[19] The superiority of CTF over other configurations can be proved only for the linear case [3].

variances) employing α–β filters was derived in [23]. For the specified case, a condition to guarantee the crosscovariance matrix's positivity was presented, which does not always hold in the heterogeneous case.

The goal of this work is to fuse the tracks from heterogeneous local sensors (an active and a passive one) with different state spaces to yield fused estimates in the full state space and evaluate the performance of the resulting heterogeneous T2TF. The fusion configuration considered is the one without memory at the FC and no feedback to the local sensors (T2TFwoMnf in the terminology of [24]).

In view of the fact that there is no known way to evaluate the crosscovariance of the estimation errors in the case of heterogeneous sensors, a Monte Carlo (MC) investigation of these errors' crosscorrelations is carried out.

The chapter is organized as follows. Section 4.2 formulates the heterogenous T2TF problem. Two approaches, namely, the linear minimum mean square error (LMMSE) and maximum likelihood (ML) heterogeneous T2TF are presented in Section 4.3. The crosscorrelation analysis by MC simulations is presented in Section 4.4. Section 4.5 evaluates the proposed approaches in a tracking scenario with an active sensor and a passive sensor. Section 4.6 provides conclusions.

4.2 The Heterogenous Fusion Problem

Without loss of generality, consider the following state-space models

$$\mathbf{x}^i(k+1) = f^i[\mathbf{x}^i(k)] + \mathbf{v}^i(k) \tag{4.2.1}$$

$$\mathbf{z}^i(k) = h^i[\mathbf{x}^i(k)] + \mathbf{w}^i(k) \tag{4.2.2}$$

at sensor i and

$$\mathbf{x}^j(k+1) = f^j[\mathbf{x}^j(k)] + \mathbf{v}^j(k) \tag{4.2.3}$$

$$\mathbf{z}^j(k) = h^j[\mathbf{x}^j(k)] + \mathbf{w}^j(k) \tag{4.2.4}$$

at sensor j. In the above, $f^s[\cdot]$ and $h^s[\cdot]$, $s = i, j$, are different and can be nonlinear; $\mathbf{v}^s(\cdot)$ and $\mathbf{w}^s(\cdot)$, $s = i, j$, are the process and measurement noises, respectively.

Further, note that $\mathbf{x}^i$ and $\mathbf{x}^j$ are in different state spaces. Let $\mathbf{x}^i$ be the larger dimension state (e.g., full Cartesian position and velocity in 2-dimensional space for tracking with an active sensor)

$$\mathbf{x}^i = [\, x \;\; \dot{x} \;\; y \;\; \dot{y} \,]' \tag{4.2.5}$$

and $\mathbf{x}^j$ be the smaller dimension state (e.g., angular position and velocity for tracking with a passive sensor)

$$\mathbf{x}^j = [\, \theta \;\; \dot{\theta} \,]' \tag{4.2.6}$$

These state vectors have the nonlinear relationship

$$\mathbf{x}^j \triangleq g(\mathbf{x}^i) \tag{4.2.7}$$

The two sensors are assumed synchronized[20] and the time index k for sampling time t_k will be omitted if there is no ambiguity.

The corresponding estimates (approximate conditional means) at these heterogeneous local sensors are $\hat{\mathbf{x}}^i$ with (conditional) covariance matrix

$$P^i \triangleq E[(\mathbf{x}^i - \hat{\mathbf{x}}^i)(\mathbf{x}^i - \hat{\mathbf{x}}^i)'] \tag{4.2.8}$$

and $\hat{\mathbf{x}}^j$ with (conditional) covariance matrix

$$P^j \triangleq E[(\mathbf{x}^j - \hat{\mathbf{x}}^j)(\mathbf{x}^j - \hat{\mathbf{x}}^j)'] \tag{4.2.9}$$

The problem is how to carry out the fusion of the estimate $\hat{\mathbf{x}}^i$ with P^i and the estimate $\hat{\mathbf{x}}^j$ with P^j to achieve a better estimation performance for the full state of interest $\mathbf{x}^i$.

4.3 Heterogenous Track-to-Track Fusion

To illustrate the effect of the crosscovariance, consider the simple homogeneous T2TF in the linear-Gaussian and symmetric case with the local track covariance

[20] Generalization to asynchronous sensors is possible [25], but the notations become very cumbersome. Without considering the crosscovariance matrix, the extension to asynchronous case is straightforward. If the crosscovariance matrix is considered (for the configuration with no memory at the FC and no information feedback to the local trackers), each track's latest estimate available at the FC is predicted to the fusion time and then they are fused using the corresponding covariance matrices.

matrices $P_S^1 = P_S^2 = P_S$ and the crosscovariance matrices $P_S^{12} = P_S^{21} = P_S^X$. The resulting fused estimate and its covariance matrix are [3]

$$\hat{\mathbf{x}}_S^F = \frac{1}{2}(\hat{\mathbf{x}}_S^1 + \hat{\mathbf{x}}_S^2) \tag{4.3.1}$$

$$P_S^F = P_S^1 - (P_S^2 - P_S^{12})(P_S^1 + P_S^2 - P_S^{12} - P_S^{21})(P_S^1 - P_S^{21}) \tag{4.3.2}$$

$$= \frac{1}{2}P_S + \frac{1}{2}P_S^X \tag{4.3.3}$$

In this case the fused estimate $\hat{\mathbf{x}}_S^F$ in (4.3.1) is independent of the crosscovariance because of the assumed symmetry. However, the corresponding covariance P_S^F in (4.3.2) has a term that depends on the crosscovariance. If $P_S^X > 0$, the fusion is optimistic if one ignores the crosscovariance (in which case the fuser calculated covariance is $\frac{1}{2}P_S$, i.e., smaller than what it should be); if $P_S^X < 0$, the fusion is pessimistic.

The crosscovariance for homogeneous fusion follows from a Lyapunov equation [3] and, consequently, it is always positive semi-definite. In the heterogeneous case while there is no known way to compute the crosscovariance matrix. As shown in the coming MC simulations, some of the crosscorrelations are positive and some are negative. They depend on the relative geometry of the two sensors and the target, as well as the target maneuvers. To further complicate the situation, the maneuvers are unknown deterministic, rather than (zero-mean white) process noise and the crosscovariance based on the process noise can be substantially different from what the maneuver causes.

The following subsections present two fusers that assume the crosscovariance is available.

4.3.1 The LMMSE Fuser

The first approach to heterogeneous T2TF is to use the linear technique based on the fundamental equations of LMMSE estimation [3]. Considering the full state estimate $\hat{\mathbf{x}}^i$ as the prediction and the smaller state estimate $\hat{\mathbf{x}}^j$ as the measurement, we have the LMMSE fused estimate

$$\hat{\mathbf{x}}^i_{\text{LMMSE}} = \hat{\mathbf{x}}^i + P_{\mathbf{xz}} P_{\mathbf{zz}}^{-1} \left[\hat{\mathbf{x}}^j - g\left(\hat{\mathbf{x}}^i\right) \right] \tag{4.3.1}$$

with the corresponding fused covariance matrix

$$P^i_{\text{LMMSE}} = P^i - P_{\mathbf{xz}} P_{\mathbf{zz}}^{-1} P'_{\mathbf{xz}} \tag{4.3.2}$$

where (as the details shown in Appendix D)

$$\begin{aligned} P_{\mathbf{xz}} &\triangleq E\left[\left(\mathbf{x}^i - \hat{\mathbf{x}}^i \right) \left(\hat{\mathbf{x}}^j - g(\hat{\mathbf{x}}^i) \right)' \right] \\ &\approx P^i (G^i)' - P^{ij} \end{aligned} \tag{4.3.3}$$

$$\begin{aligned} P_{\mathbf{zz}} &\triangleq E\left[\left(\hat{\mathbf{x}}^j - g(\hat{\mathbf{x}}^i) \right) \left(\hat{\mathbf{x}}^j - g(\hat{\mathbf{x}}^i) \right)' \right] \\ &\approx P^j - G^i P^{ij} - P^{ji} (G^i)' + G^i P^i (G^i)' \end{aligned} \tag{4.3.4}$$

with G^i the Jacobian of $g(\mathbf{x}^i)$

$$G^i \triangleq \left[\nabla_{\mathbf{x}^i} g(\mathbf{x}^i)' \right]'_{\mathbf{x}^i = \hat{\mathbf{x}}^i} \tag{4.3.5}$$

and P^{ij} the crosscovariance matrix

$$P^{ij} \triangleq E[(\mathbf{x}^i - \hat{\mathbf{x}}^i)(\mathbf{x}^j - \hat{\mathbf{x}}^j)'] \tag{4.3.6}$$

4.3.2 The ML Fuser

Under the Gaussian assumption, the heterogeneous T2TF problem can be solved by minimizing the negative log-likelihood function[21]

$$L = -\ln p(\hat{\mathbf{x}}^i, \hat{\mathbf{x}}^j | \mathbf{x}^i) \propto \left(\begin{bmatrix} \hat{\mathbf{x}}^i \\ \hat{\mathbf{x}}^j \end{bmatrix} - \begin{bmatrix} \mathbf{x}^i \\ \mathbf{x}^j \end{bmatrix} \right)' P^{-1} \left(\begin{bmatrix} \hat{\mathbf{x}}^i \\ \hat{\mathbf{x}}^j \end{bmatrix} - \begin{bmatrix} \mathbf{x}^i \\ \mathbf{x}^j \end{bmatrix} \right) \tag{4.3.1}$$

where (4.2.7) has been used and

$$P = \begin{bmatrix} P^i & P^{ij} \\ P^{ji} & P^j \end{bmatrix} \tag{4.3.2}$$

Then the ML fused estimate is the solution of

$$\nabla_{\mathbf{x}^i} L = 0 \tag{4.3.3}$$

Because of the nonlinearity of the function $g(\mathbf{x}^i)$, there is no explicit expression for the solution of (4.3.3). It can be solved by a numerical search, e.g., the gradient projection algorithm. The result is denoted as $\hat{\mathbf{x}}^i_{\mathrm{ML}}$ and the corresponding

[21] As it is pointed out in [5], the LMMSE T2TF approach is, in the linear Gaussian case, optimal in ML sense.

covariance matrix is

$$P^i_{\mathrm{ML}} = \left(\begin{bmatrix} IG^i \end{bmatrix} P^{-1} \begin{bmatrix} I \\ G^i \end{bmatrix} \right)^{-1} \tag{4.3.4}$$

where G^i is defined in (4.3.5) and I is the identity matrix (4×4 in our case).

The results of $\hat{\mathbf{x}}^i_{\mathrm{LMMSE}}$ with P^i_{LMMSE} and $\hat{\mathbf{x}}^i_{\mathrm{ML}}$ with P^i_{ML} will be examined and compared with the CTF which processes all the measurements (from both the active and the passive sensor) in the FC in the simulation section.

4.4 Crosscorrelation in Heterogeneous Fusion

It has been recognized that, although different local sensors typically have independent measurement noises, the process noises for the motion models at these sensors belong to the same target and, consequently, will lead to (cross)correlated state estimation errors. This is the so-called "common process noise effect" [3]. For the heterogenous case, the common process noise effect, as it is embedded into the estimates from different sensors for the same target, also exists. However, since the estimates are in different state spaces, there is no known way to capture the "common" part exactly.

The dependence of the estimation errors can be quantified by the crosscovariance matrix, and the more accurately the crosscovariance matrix is obtained, the better the heterogeneous track-to-track fusion performance will be. However, the difference between the motion models for different sensors prohibits

the evaluation of the crosscovariance matrix by the exact method described in [3] (limited to the homogeneous case and linear systems). Even this exact method is not considered practical since it requires information that is typically not available at the FC (the local filter gains).

While process noise is used in the motion equations to model the target maneuvers[22], these maneuvers are, however, not stochastic process. Consequently, MC simulations will be used to evaluate the crosscorrelation of the estimation errors from different sensors. As shown in Appendix F, considering the estimates from different local sensors in each MC run as one sample and evaluating the sample crosscorrelation coefficients, we observe both negative and positive crosscorrelations of the estimation errors from the heterogeneous local sensors in different parts of the target trajectory.

The fact that these crosscorrelations can be, unlike in the linear homogeneous case (when they are always positive), sometimes positive and sometimes negative is shown as follows. According to (4.3.2), the information matrix (assuming P^{ij} is available) is

$$\begin{aligned} J=&(P^F)^{-1} \\ =&\Big[P^i-[P^i(G^i)'-P^{ij}][P^j-G^iP^{ij}-P^{ji}(G^i)'+G^iP^i(G^i)']^{-1}[P^i(G^i)'-P^{ij}]'\Big]^{-1} \\ \triangleq&\Big[P^i-[P^i(G^i)'-P^{ij}][P^j+G^iP^i(G^i)'+U]^{-1}[P^i(G^i)'-P^{ij}]'\Big]^{-1} \end{aligned} \tag{4.4.1}$$

[22]The whiteness is necessary so the state is a Markov process, a sine qua non prerequisite for any recursive estimation algorithm [2].

where

$$U \triangleq -G^i P^{ij} - P^{ji}(G^i)' \tag{4.4.2}$$

Assuming $P^{ij} = 0$ (its elements are all zero), designated as the "uncorrelated" assumption (denoted concisely as "uncorr"), then (4.4.1) can be simplified (by the matrix inversion lemma) as

$$\begin{aligned} J(P^{ij}=0) &= \left[P^i - P^i(G^i)'[P^j + G^i P^i (G^i)']^{-1} G^i (P^i)'\right]^{-1} \\ &= (P^i)^{-1} - [(P^i)^{-1} P^i (G^i)'] \\ &\quad \cdot [G^i (P^i)' (P^i)^{-1} P^i (G^i)' - P^j - G^i P^i (G^i)']^{-1} \\ &\quad \cdot [(P^i)^{-1} P^i (G^i)']' \\ &= (P^i)^{-1} + (G^i)'(P^j)^{-1} G^i \end{aligned} \tag{4.4.3}$$

If $P^{ij} \neq 0$ (this is denoted as "corr" for conciseness), then we have

$$\begin{aligned} J(P^{ij} \neq 0) &= (P^i)^{-1} - [(G^i)' - (P^i)^{-1} P^{ij}] \\ &\quad \cdot \left[[P^i (G^i)' - P^{ij}]' (P^i)^{-1} [P^i (G^i)' - P^{ij}] - [P^j + G^i P^i (G^i)' + U]\right]^{-1} \\ &\quad \cdot [(G^i)' - (P^i)^{-1} P^{ij}]' \\ &\triangleq (P^i)^{-1} + [(G^i)' - (P^i)^{-1} P^{ij}][P^j + W]^{-1} [(G^i)' - (P^i)^{-1} P^{ij}]' \end{aligned} \tag{4.4.4}$$

where

$$W\stackrel{\Delta}{=}[G^iP^i(G^i)' + U] - [P^i(G^i)' - P^{ij}]'(P^i)^{-1}[P^i(G^i)' - P^{ij}]$$
$$=-P^{ji}(P^i)^{-1}P^{ij} \tag{4.4.5}$$

Equation (4.4.4) can be written as (the generic matrix inversion lemma is used)

$$\begin{aligned} J(P^{ij} \neq 0)=&(P^i)^{-1} + [(G^i)' - (P^i)^{-1}P^{ij}] \\ &\cdot[(P^j)^{-1} - (P^j)^{-1}W(P^j)^{-1}[I + W(P^j)^{-1}]^{-1}] \\ &\cdot[(G^i)' - (P^i)^{-1}P^{ij}]' \\ \stackrel{\Delta}{=}&(P^i)^{-1} + [(G^i)' - (P^i)^{-1}P^{ij}][(P^j)^{-1} - K][(G^i)' - (P^i)^{-1}P^{ij}]' \\ \stackrel{\Delta}{=}&(P^i)^{-1} + (G^i)'(P^j)^{-1}G^i - K_c \\ =&J(P^{ij} = 0) - K_c \end{aligned} \tag{4.4.6}$$

where I is the identity matrix and

$$K\stackrel{\Delta}{=}(P^j)^{-1}W(P^j)^{-1}[I + W(P^j)^{-1}]^{-1} \tag{4.4.7}$$

$$\begin{aligned} K_c\stackrel{\Delta}{=}&[(G^i)' - (P^i)^{-1}P^{ij}]K[(G^i)' - (P^i)^{-1}P^{ij}]' + (P^i)^{-1}P^{ij}(P^j)^{-1}G^i \\ &+ [(P^i)^{-1}P^{ij}(P^j)^{-1}(G^i)]' - [(P^i)^{-1}P^{ij}](P^j)^{-1}[(P^i)^{-1}P^{ij}]' \end{aligned} \tag{4.4.8}$$

Setting $P^{ij} = 0$ and $P^{ij} \neq 0$ for the estimation from the heterogenous local sensors correspond to "uncorr" and "corr" assumptions, respectively. For the homogeneous case, the crosscovariance matrix is always positive; this follows

from the Lyapunov equation (9.3.2-3) in [3]. However, in the heterogeneous (and nonlinear) case, K_c may be indefinite (some eigenvalues are positive and some negative). Therefore, accounting for it (as opposed to assuming it zero) yields the exact (optimal) variance larger in some state components and smaller in others.

The results using a simple functional model of the crosscorrelation of the estimation errors ($\tilde{\mathbf{x}}^i$ and $\tilde{\mathbf{x}}^j$), based on the polar-to-Cartesian transformation, are shown in Appendix G and they do not provide any perceivable benefits. As discussed above, for the linear-Gaussian and symmetric case, neglecting the common process noise makes the fusion optimistic. For the nonlinear case examined, neglecting it makes the fusion sometimes optimistic and sometimes pessimistic, but the effect is small. This supports the approach of ignoring the dependency between the tracks from different local sensors.[23] Thus, since the maneuvers are unknown and scenario dependent, we pursue the heterogeneous T2TF without considering the crosscorrelation between the estimation errors.

4.5 Simulation Results

A typical 2-dimensional scenario for heterogenous T2TF is with an active sensor

[23] We are grateful for the anonymous reviewer who summarized so well our findings.

located at $(x_a,\ y_a)$, with measurements of target range and azimuth angle

$$r=\sqrt{(x-x_a)^2+(y-y_a)^2}+w_r \tag{4.5.1}$$

$$\theta_a=\tan^{-1}\left(\frac{y-y_a}{x-x_a}\right)+w_a \tag{4.5.2}$$

and a passive sensor located at $\left(x_p,\ y_p\right)$, with the azimuth angle measurements

$$\theta_p=\tan^{-1}\left(\frac{y-y_p}{x-x_p}\right)+w_p \tag{4.5.3}$$

where w_r, w_a and w_p are mutually independent zero mean white Gaussian noises with standard deviations (SD) σ_r, σ_a and σ_p, respectively.

The ground truth is a target moving with a constant speed of 250 m/s with initial state in Cartesian coordinates (with position in m)

$$\mathbf{x}^i(0)=[\,x(0)\ \ \dot{x}(0)\ \ y(0)\ \ \dot{y}(0)\,]'=[\,0\ \ 0\ \ 20000\ \ 250\,]' \tag{4.5.4}$$

At $t=100$ s it starts a left turn of 2°/s (about 30 mrad/s) for 30 s, then continues straight until $t=200$ s, at which time it turns right with 1°/s for 45 s, then left with 1°/s for 90 s, then right with 1°/s for 45 s, then continues straight until 50 s.

The measurements of the active sensor located at $(-6\cdot10^4,\ 2\cdot10^4)$ m are made every $T_a=5$ s, starting from $k=0$ with measurement noise SD as $\sigma_r=20$ m and $\sigma_a=5$ mrad. An unbiased measurement conversion from polar coordinates to Cartesian coordinates (denoted as $\mathbf{z}^i$ with $\mathbf{w}^i$ for polar coordinates and $\mathbf{z}_c^i$ with $\mathbf{w}_c^i$ for Cartesian coordinates, respectively) is done for the active sensor

measurements for filtering [3]. The measurements of the passive sensor located at $(-5 \cdot 10^4,\ 4 \cdot 10^4)$ m are made every $T_p = 1$ s, starting from $k = 0$ with measurement noise $\sigma_p = 1$ mrad. This scenario is shown in Fig.4.5.1.

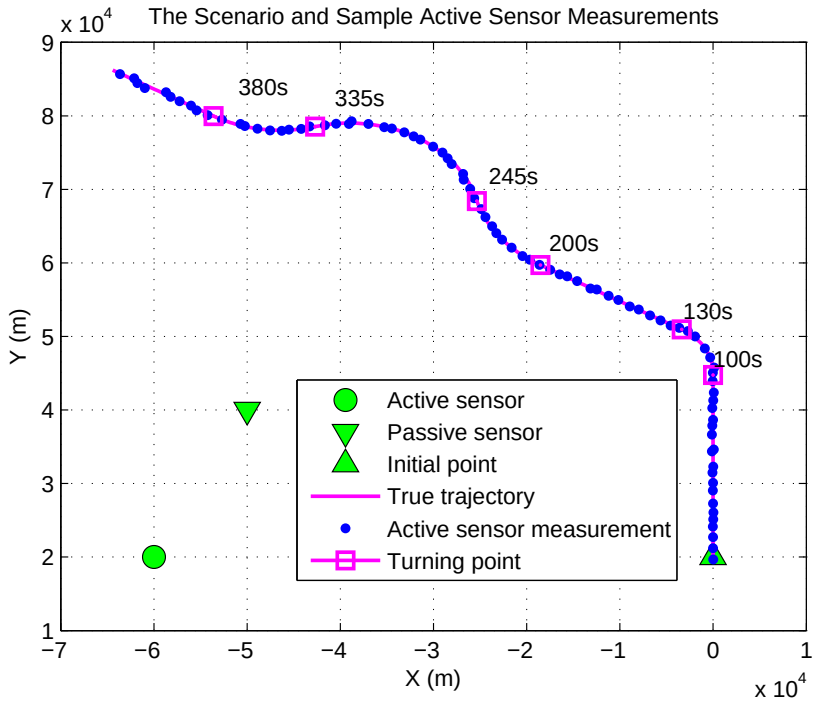


Figure 4.5.1: The scenario.

The active sensor uses an interacting multiple model (IMM) estimator using continuous time white noise acceleration (CWNA) model [3]. The passive sensor uses, for reasons shown in Appendix H, a linear Kalman filter (KF) using a continuous time Wiener process acceleration (CWPA) model [3].

4.5.1 Active Sensor IMM Estimator Design

For the active sensor IMM, in order to cover the uniform motion segments and maneuvering segments in the trajectory, two modes are used: one mode using a linear nearly constant velocity (NCV) model that includes 4 state components

(the Cartesian position and velocity in X- and Y- coordinates) and the other mode using a nearly coordinated turn (NCT) model that includes that 5 state components (the Cartesian position and velocity in X- and Y- coordinates and the turn rate Ω)

$$\mathbf{x}_a^i(k) \triangleq [\ x(k) \quad \dot{x}(k) \quad y(k) \quad \dot{y}(k) \quad \Omega(k)\ ]' = [\ \mathbf{x}^i(k)' \quad \Omega(k)\ ]' \tag{4.5.1}$$

Since the conventional mixing [3] in an IMM estimator with modes that have unequal dimension state vectors will lead to bias in the estimates of the non-common state component(s), the unbiased mixing approach described in [27] is used to overcome this bias problem (for the turn rate in this case).

The NCV model for uniform motion segment is implemented as the following discretized CWNA model [3] (with low-level process noise)

$$\mathbf{x}^i(k+1) = F^i(k)\mathbf{x}^i(k) + \mathbf{v}^i(k) \tag{4.5.2}$$

$$\mathbf{z}_c^i(k) = H^i(k)\mathbf{x}^i(k) + \mathbf{w}_c^i(k) \tag{4.5.3}$$

where, with T_a is the sampling interval of the active sensor [24],

$$F^i = \begin{bmatrix} 1 T_a 0 0 \\ 0 1 0 0 \\ 0 0 1 T_a \\ 0 0 0 1 \end{bmatrix} \tag{4.5.4}$$

$$H^i = \begin{bmatrix} 1000 \\ 0010 \end{bmatrix} \tag{4.5.5}$$

and the covariance matrix of the process noise $\mathbf{v}^i(k)$ is

$$Q^i(k) \triangleq E\left[\mathbf{v}^i(k)\mathbf{v}^i(k)'\right] = \begin{bmatrix} \begin{bmatrix} \frac{T_a^3}{3} & \frac{T_a^2}{2} \\ \frac{T_a^2}{2} & T_a \end{bmatrix} q_a & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \begin{bmatrix} \frac{T_a^3}{3} & \frac{T_a^2}{2} \\ \frac{T_a^2}{2} & T_a \end{bmatrix} q_a \end{bmatrix} \tag{4.5.6}$$

where the continuous time process noise "intensities" q_a is the power spectral density (PSD). Note that the process noise induced root mean square (RMS) change in the velocity over a sampling interval T_a is

$$d_a \triangleq \frac{\sqrt{q_a T_a}}{T_a} \tag{4.5.7}$$

whose physical dimensions is linear acceleration [31].

The NCT model, commonly refers to a target maneuver executed under nearly

[24] Different sampling rates may be possible and T_a is replaced by the time-variant $T_a(k) \triangleq t(k+1) - t(k)$.

constant speed along a nearly circular path, is implemented as the following discretized continuous-time coordinated turn dynamic model [16]

$$\mathbf{x}_a^i(k+1) = f_a^i[\mathbf{x}_a^i(k)] + \mathbf{v}_a^i[\mathbf{x}_a^i(k)] \tag{4.5.8}$$

$$\mathbf{z}_c^i(k) = H_a^i \mathbf{x}_a^i(k) + \mathbf{w}_c^i(k) \tag{4.5.9}$$

where

$$f_a^i[\mathbf{x}_a^i(k)] = \begin{bmatrix} x(k) + T_a\dot{x}(k) - T_a^2\Omega(k)\dot{y}(k)/2 \\ \dot{x}(k) - T_a\Omega(k)\dot{y}(k) - T_a^2\Omega(k)^2\dot{x}(k)/2 \\ y(k) + T_a\dot{y}(k) + T_a^2\Omega(k)\dot{x}(k)/2 \\ \dot{y}(k) + T_a\Omega(k)\dot{x}(k) - T_a^2\Omega(k)^2\dot{y}(k)/2 \\ \Omega(k) \end{bmatrix} \tag{4.5.10}$$

$$H_a^i = \begin{bmatrix} 10000 \\ 00100 \end{bmatrix} \tag{4.5.11}$$

and the process noise for the NCT model depends on current target state (target velocity components and turn rate). The covariance matrix of the target state-dependent process noise $\mathbf{v}_a^i[\mathbf{x}_a^i(k)]$ can be roughly shown as (the detailed form

can be found in Appendix E) [25]

$$Q_a^i[\mathbf{x}_a^i(k)] \triangleq E\left\{\mathbf{v}_a^i[\mathbf{x}_a^i(k)]\mathbf{v}_a^i[\mathbf{x}_a^i(k)]'\right\} = \begin{bmatrix} \frac{T_a^3}{3}\frac{\dot{x}(k)^2}{\dot{x}(k)^2+\dot{y}(k)^2}q_a & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \frac{T_a^3}{3}\frac{\dot{y}(k)^2}{\dot{x}(k)^2+\dot{y}(k)^2}q_a & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & Tq_\Omega \end{bmatrix} \tag{4.5.12}$$

where the continuous time process noise "intensities" q_a and q_Ω are the PSDs. Note that the process noise induced RMS change over a sampling interval T_a in the velocity is as in (4.5.7) and in the turn rate is

$$d_\Omega \triangleq \frac{\sqrt{q_\Omega T_a}}{T_a} \tag{4.5.13}$$

whose physical dimensions is turn acceleration [31]. The d_a and d_Ω are the design values used to select the process noise PSDs. A guideline for the choice of these process noise intensities for the NCT model is shown in Appendix E.

As the NCT model described in (4.5.8) is nonlinear, extended KF (EKF) has been used as the mode-matched filter for the NCT model in active sensor IMM. We only use the estimate $\hat{\mathbf{x}}^i(k)$ (from $\hat{\mathbf{x}}_a^i(k)$) and the corresponding covariance matrix $P^i(k)$ for the fusion.

For the active sensor IMM estimator with unbiased converted measurements

[25] This "target state-dependent" process noise covariance matrix from [16] yielded superior RMSE performance (but no major change in the consistency) compared with the simplified covariance matrix from [3], Sec. 11. 7.

(in Cartesian coordinates), the process noises design values are summarized in Table 4.5.18 and the corresponding transition probability matrix is (based on the mean sojourn time [3])

$$\pi_{CT} = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix} \tag{4.5.14}$$

with initial mode probability vector [0.9, 0.1].

Table 4.5.18: RMS change rate due to process noise (NCV and NCT)

	d_a (m/s^2)	d_Ω (mrad/s^2)
Mode 1 (NCV)	0.2	N/A
Mode 2 (NCT)	1	2

4.5.2 Passive Sensor KF Estimator Design

For the passive sensor, as pointed out in Appendix C, the target maneuvering index is very small and the target maneuvers are nearly unobservable by the passive sensor. Consequently, a single model filter (i.e., a linear KF) has been chosen as the local estimator, with the state vector

$$\mathbf{x}_p^j \triangleq [\, \theta \quad \dot{\theta} \quad \ddot{\theta} \,]' = [\, \mathbf{x}^j(k)' \quad \ddot{\theta} \,]' \tag{4.5.1}$$

The discretized CWPA model [3] in the angle, angle rate and angle acceleration

space is

$$\mathbf{x}_p^j(k+1)=F_p^j\mathbf{x}_p^j(k)+\mathbf{v}_p^j(k) \tag{4.5.2}$$

$$\mathbf{z}_p^j(k)=H_p^j\mathbf{x}_p^j(k)+\mathbf{w}_p^j(k) \tag{4.5.3}$$

where, with T_p the sampling interval of the passive sensor,

$$F_p^j=\begin{bmatrix} 1 & T_p & \frac{T_p^2}{2} \\ 0 & 1 & T_p \\ 0 & 0 & 1 \end{bmatrix} \tag{4.5.4}$$

$$H_p^j=\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{4.5.5}$$

$$\tag{4.5.6}$$

and the covariance matrix of the process noise is

$$Q_p^j(k) \triangleq E\left[\mathbf{v}_p^j(k)\mathbf{v}_p^j(k)'\right]=\begin{bmatrix} \frac{T_p^5}{20} & \frac{T_p^4}{8} & \frac{T_p^3}{6} \\ \frac{T_p^4}{8} & \frac{T_p^3}{3} & \frac{T_p^2}{2} \\ \frac{T_p^3}{6} & \frac{T_p^2}{2} & T_p \end{bmatrix} q_p \tag{4.5.7}$$

Note that for the PSD q_p, the process noise induced RMS change in the angular acceleration over T_p is

$$d_p \triangleq \frac{\sqrt{q_p T_p}}{T_p} \tag{4.5.8}$$

whose physical dimension is angular jerk (derivative of acceleration).

The process noise design value chosen for the passive sensor is $d_p = 0.04$ (mrad/s^3). We only use the estimate $\hat{\mathbf{x}}^j(k)$ (from $\hat{\mathbf{x}}_p^j(k)$) and the corresponding covariance matrix $P^j(k)$ for the fusion.

4.5.3 Results of Heterogeneous T2TF

The LMMSE and ML heterogeneous T2TF are carried out at the FC every $T_f = 5\,\text{s}$ under the "uncorr" assumption, with the local estimates $\hat{\mathbf{x}}^i(k)$ (from $\hat{\mathbf{x}}_a^i(k)$) and $\hat{\mathbf{x}}^j(k)$ (from $\hat{\mathbf{x}}_p^j(k)$) and their corresponding covariance matrices $P^i(k)$ and $P^j(k)$. The CTF uses the same IMM design (CTF IMM for short) as the active sensor IMM estimator. The FC can run the fusion at an arbitrarily low rate or "on demand".

Results of LMMSE Fuser

In Figs. 4.5.3 and 4.5.4, the root mean square errors (RMSE) for the LMMSE fuser (with $T_f = 5\,\text{s}$ under the "uncorr" assumption) are compared with those for the active sensor's IMM estimator and the CTF IMM in position and velocity, respectively. It can be seen that the LMMSE heterogeneous T2TF approach always provides significantly better estimation performance than the single (active) sensor case.

The LMMSE heterogeneous T2TF provides larger RMSE than the CTF IMM in the non-maneuvering intervals but smaller RMSE if there is a maneuver. This degradation of the CTF in both position and velocity during the maneuvering intervals is because the CTF is using an IMM estimator, which is inappropriate

for the passive sensor (due to the very small maneuver index). While using the IMM estimator is generally beneficial for maneuvering targets, the use of an IMM estimator with a sensor that cannot "see" the maneuvers can lead to performance degradation (the CTF IMM's performance at some fusion points is even worse than the active sensor IMM's). As shown in Appendix H, the maneuvering index from the passive sensor's view is so small that when the passive sensor measurements (with higher sampling rate than those of the active sensor) are sent to FC and processed centrally, these measurements increase the uncertainty about the target maneuvers.

From the maneuvering mode probability (NCT) in the active sensor IMM and in the CTF IMM, shown in Fig. 4.5.2, it can be seen that the CTF IMM can not "see" the maneuvers at the times when there is only a passive sensor measurement and its maneuvering mode probability becomes too small. The use of the passive measurements in the CTF IMM "clouds" the maneuvers – it is preferable to have an active sensor IMM (which does detect the maneuvers) and a passive sensor KF (since the passive sensor is almost "blind" to the maneuvers) and fuse the outputs of these two local trackers.

The observation from Figs. 4.5.3 and 4.5.4 that the CTF IMM performs during target maneuvers worse than the heterogeneous T2TF points out that the heterogeneous T2TF benefits from the freedom of having more suitable filters for the individual local sensors. This freedom can provide final fusion results comparable or even better than the corresponding CTF estimator.

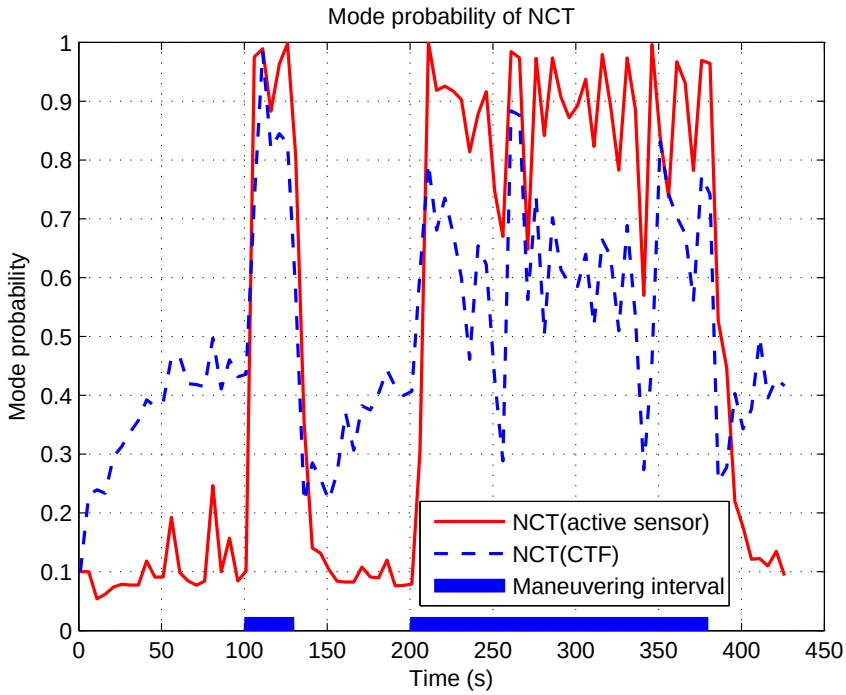


Figure 4.5.2: Probability of maneuvering mode (NCT) in active sensor IMM and CTF IMM

We evaluate the fusion consistency of the LMMSE fuser by the normalized estimation error square (NEES) consistency test [3]. The NEES for the LMMSE fusion approach are shown in Fig. 4.5.5. The reason for the inconsistency of the fused estimates are (i) the local IMM estimator (for the active sensor) and the KF estimator (for the passive sensor) are not entirely consistent[26] (as shown in Appendix I) and (ii) the crosscovariance has been assumed zero. Nevertheless, the quality of the estimates is improved by fusion, which justifies the approach. At this point, there is no known way to improve the sometimes optimistic, sometimes pessimistic behavior of the IMM — it is the inconsistency that

[26]The IMM estimator is the worst estimator in terms of consistency except for all the other estimators [3]. However, it is the "short term" inconsistency that is the key for the capability of the IMM estimator to adapt itself to the observed behavior of the target (large innovations).

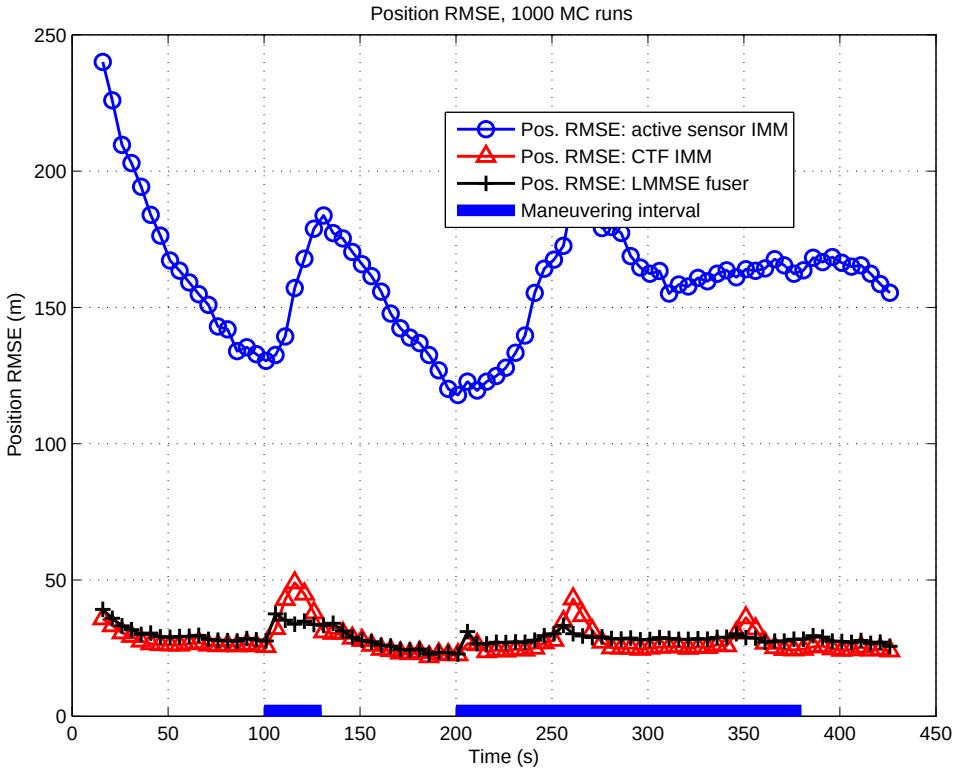


Figure 4.5.3: Position RMSE for LMMSE fuser.

drives its adaptation.

Results of ML Fuser

Using a numerical search (the gradient projection algorithm), the RMSE in position and velocity for the ML fuser are shown (with $T_f = 5\,\mathrm{s}$ under the "uncorr" assumption) in Fig. 4.5.6 and Fig. 4.5.7, respectively. It can be seen that both the LMMSE fuser and the ML fuser give practically the same RMSE in position and velocity and both have better performances than the single (active) sensor case. As pointed out in [5], the LMMSE fuser is, in the linear-Gaussian case, actually optimal in the ML sense. Since the ML fuser in the heterogenous case (with nonlinearity) needs to be implemented by a time-

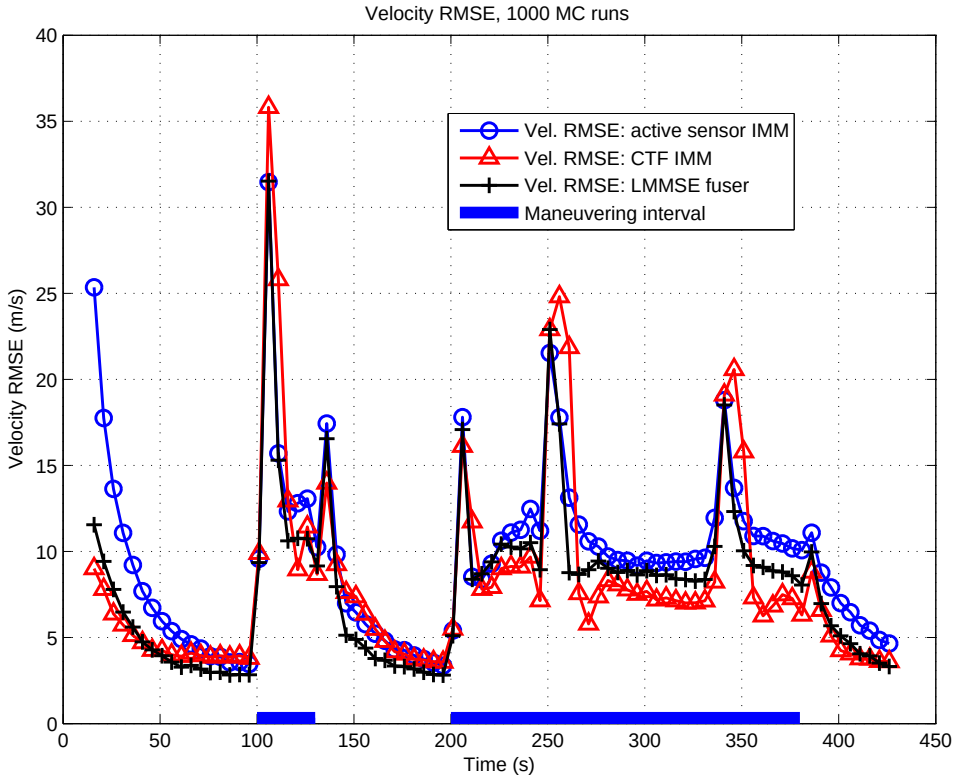


Figure 4.5.4: Velocity RMSE for LMMSE fuser.

consuming numerical search, the LMMSE fuser can be considered as an efficient and effective alternative for the ML fuser.

4.6 Conclusions

Examining the differences between the heterogeneous and homogeneous T2TF, this chapter investigates the major difficulties of heterogenous T2TF. The LMMSE and the ML approaches for heterogenous T2TF are presented and compared with the corresponding CTF. The simulation study shows that both approaches can effectively achieve improved performance over the single sensor track quality and comparable performance to the CTF track. The use of the passive

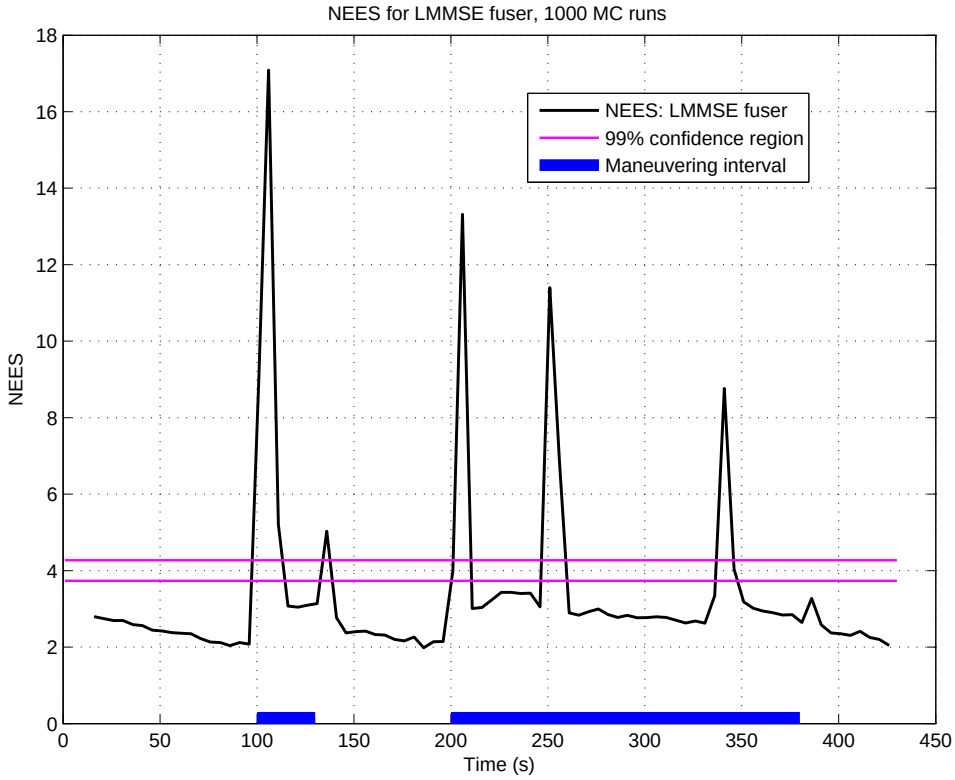


Figure 4.5.5: NEES for LMMSE fuser.

measurements in the CTF IMM "clouds" the maneuvers – it is preferable to have an active sensor IMM (which does detect the maneuvers) and a passive sensor KF (since the passive sensor is almost "blind" to the maneuvers) and fuse the outputs of these two local trackers. The freedom available to each local sensor to flexibly design a more suitable local estimator allows the heterogeneous T2TF approach to achieve a better estimation performance than the CTF IMM in the scenario considered. As the LMMSE T2TF has practically the same performance as the ML T2TF, it can be considered as an effective and efficient alternative for the numerical search required by the ML approach. The estimation errors' crosscorrelation has been examined by MC simulations. As

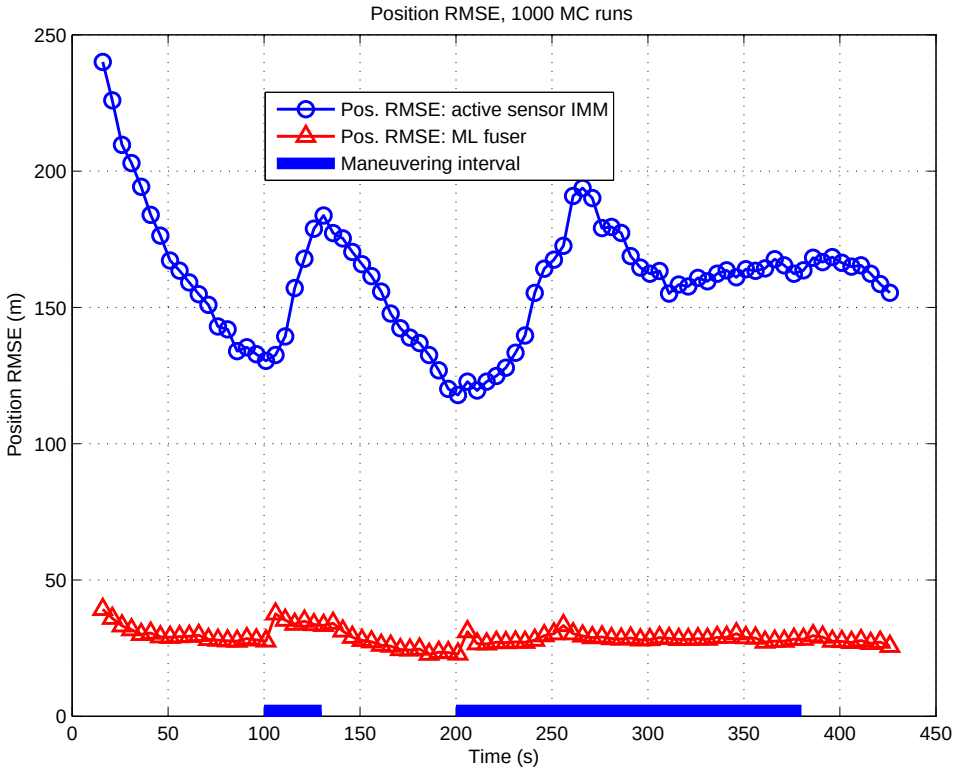


Figure 4.5.6: Position RMSE for ML fuser.

it is impossible to predict maneuvers in a trajectory and there is no known way to correctly quantify the crosscorrelation of the estimation errors from heterogeneous local sensors, the heterogeneous T2TF was carried out assuming the tracks from the heterogeneous local sensors as uncorrelated.

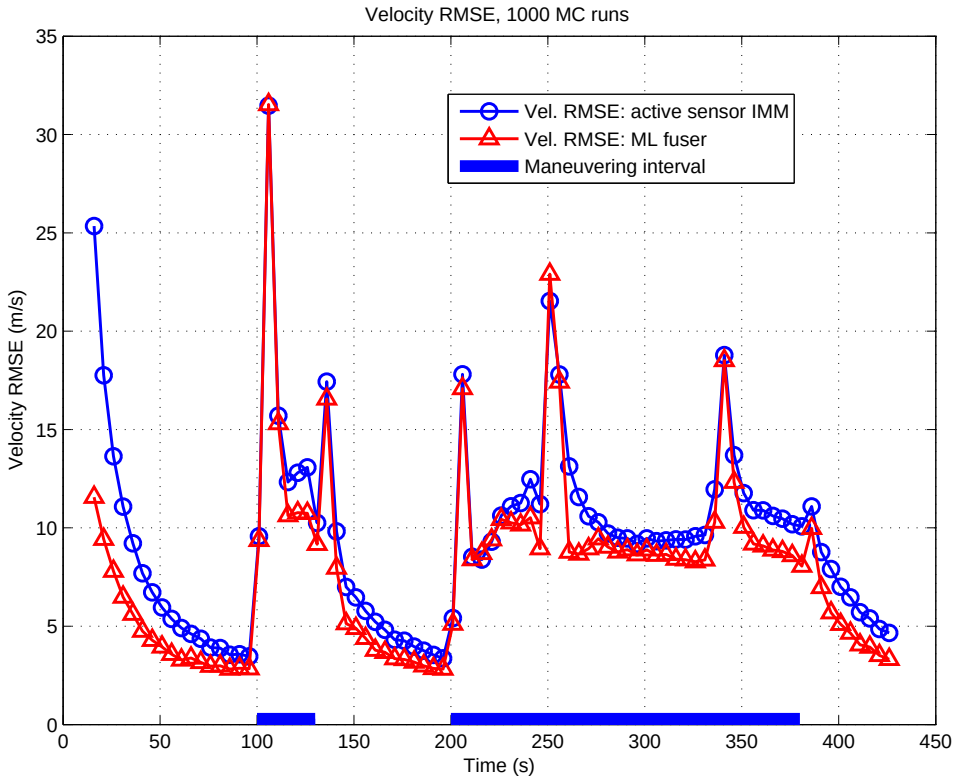


Figure 4.5.7: Velocity RMSE for ML fuser.

Chapter 5
Conclusions

In Chapter 1, we presented a target state estimation approach for IPP using full-position measurements. The proposed MIMM estimator with a grid of initial drag coefficient estimates largely mitigates the estimation ambiguity between the critical drag coefficient and thrust components. The novel unbiased mixing is the key for correct (unbiased) estimation of the extra thrust component and, consequently, it is conducive to more accurate estimates of the drag coefficient. The MIMM estimator with the unbiased mixing (MIMM-UM) significantly alleviates the uncertainty in the motion model and the drag-thrust estimation ambiguity and yields the final IPP of good accuracy.

In Chapter 2, we extended the MIMM-UM approach for IPP to a more complicated environment, namely, in the presence of wind. The system model was modified to account for the wind effect. We presented an elaborate design (algorithm development and parameter selection) and an extensive analysis of the wind effect on the IPP system. A MC simulation study was conducted based on various assumptions on the knowledge of the wind information: given no wind information, given perfect wind information and given wind information with a deterministic bias. We showed that, if the wind information

is available, the wind effect can be fully compensated. Otherwise, the short observation time and the limited sensor accuracy are more critical since they could cause confusion in selecting the "correct" best IMM estimator from the MIMM estimator, and lead to a degraded IPP performance. Then increasing the total observation time, which can mitigate the wind effect, and employing the proposed N-point initialization, which can speed up the correct mode convergence in each IMM estimator, become desirable.

In Chapter 3, a target parameter estimation approach for IPP using LoS measurements from a single passive sensor was presented. The simpler case with the known LP altitude was analyzed first and then it was considered as an unknown (additional) target parameter to estimate. As the possible existence of ambiguity in estimating a 3-D trajectory from 2-D measurements, a full-rank Fisher information matrix (FIM) is required for estimable target parameters whether or not the LP altitude is to be estimated. With the FIM shown to be full-rank, the corresponding CRLB was obtained and used for the IPP accuracy evaluation. In view of the inherent nonlinearity of the problem, the maximum likelihood (ML) estimate of the target parameter vector was sought. A search strategy with two stages — a mixed (partially grid-based) search followed by a continuous search — was proposed. According to a simulation study, the approach was shown to have reliable estimation performance and leading to a final IPP of good accuracy. Furthermore, due to its parallelizable nature, the mixed search allows the two-stage strategy to be real-time implementable.

In Chapter 4, the heterogenous track-to-track fusion (HT2TF) problem involving the fusion of tracks in different state spaces was discussed. We observed that, for this nonlinear HT2TF system, the crosscorrelation of the estimation errors from heterogeneous local trackers is too complicated to capture — it can be positive or negative. The estimation errors' crosscorrelation was examined by MC simulations. When using a LMMSE/ML fuser, neglecting the track crosscovariance in HT2TF leads to sometimes optimistic, sometimes pessimistic fused covariance. This is different from the homogenous T2TF where neglecting the crosscovariance always results in optimistic fused covariance (for linear systems the crosscorrelation coefficients are always positive). However, when the configuration of fusion without memory and no information feedback is used, neglecting the track crosscovariance is a reasonable practical choice and yields little loss in fusion performance. In a simulation study for a maneuvering target, we observed the use of the passive measurements in the CTF IMM "clouds" the maneuvers because of low maneuvering index at the passive sensor and leads to the degraded performance of the CTF IMM filter. The HT2TF allows each local sensor to flexibly design a more suitable local estimator which was shown to lead to a better estimation performance than the CTF, unlike in the linear homogeneous case.

Appendix A

Covariance Matrix of Higher Dimensional Mode in Unbiased Mixing

The covariance matrix of the higher dimensional mode 1 of the unbiased mixing is ($O^d(k), d \in \{1,2\}$ is the possible mode of mode-matched filter d at time k)

$$
\begin{aligned}
P_M^{01} &\triangleq E\left((\mathbf{x}-\hat{\mathbf{x}}_M^{01})(\mathbf{x}-\hat{\mathbf{x}}_M^{01})'|O^1(k),Z^{k-1}\right) \\
&= E\left[(\mathbf{x}-\hat{\mathbf{x}}^1+\hat{\mathbf{x}}^1-\hat{\mathbf{x}}_M^{01})(\mathbf{x}-\hat{\mathbf{x}}^1+\hat{\mathbf{x}}^1-\hat{\mathbf{x}}_M^{01})'|O^1(k-1),O^1(k),Z^{k-1}\right]\mu_{1|1} \\
&\quad + E\left[(\mathbf{x}-\hat{\mathbf{x}}_M^2+\hat{\mathbf{x}}_M^2-\hat{\mathbf{x}}_M^{01})(\mathbf{x}-\hat{\mathbf{x}}_M^2+\hat{\mathbf{x}}_M^2-\hat{\mathbf{x}}_M^{01})'|O^2(k-1),O^1(k),Z^{k-1}\right]\mu_{2|1} \\
&= \left\{E\left[(\mathbf{x}-\hat{\mathbf{x}}^1)(\mathbf{x}-\hat{\mathbf{x}}^1)'|O^1(k-1),O^1(k),Z^{k-1}\right]+(\hat{\mathbf{x}}^1-\hat{\mathbf{x}}_M^{01})(\hat{\mathbf{x}}^1-\hat{\mathbf{x}}_M^{01})'\right\}\mu_{1|1} \\
&\quad \left\{E\left[(\mathbf{x}-\hat{\mathbf{x}}_M^2)(\mathbf{x}-\hat{\mathbf{x}}_M^2)'|O^2(k-1),O^1(k),Z^{k-1}\right]+(\hat{\mathbf{x}}_M^2-\hat{\mathbf{x}}_M^{01})(\hat{\mathbf{x}}_M^2-\hat{\mathbf{x}}_M^{01})'\right\}\mu_{2|1} \\
&= \left\{\begin{bmatrix} P_c^1 & P_{ce}^1 \\ P_{ec}^1 & P_e^1 \end{bmatrix}+\left(\begin{bmatrix}\hat{\mathbf{x}}_c^1\\ \hat{\mathbf{x}}_e^1\end{bmatrix}-\begin{bmatrix}\hat{\mathbf{x}}_c^{01}\\ \hat{\mathbf{x}}_{Me}^{01}\end{bmatrix}\right)\left(\begin{bmatrix}\hat{\mathbf{x}}_c^1\\ \hat{\mathbf{x}}_e^1\end{bmatrix}-\begin{bmatrix}\hat{\mathbf{x}}_c^{01}\\ \hat{\mathbf{x}}_{Me}^{01}\end{bmatrix}\right)'\right\}\mu_{1|1} \\
&\quad + \left\{E\left[\left(\begin{bmatrix}\mathbf{x}_c\\ \mathbf{x}_e\end{bmatrix}-\begin{bmatrix}\hat{\mathbf{x}}_c^2\\ \hat{\mathbf{x}}_e^1\end{bmatrix}\right)\left(\begin{bmatrix}\mathbf{x}_c\\ \mathbf{x}_e\end{bmatrix}-\begin{bmatrix}\hat{\mathbf{x}}_c^2\\ \hat{\mathbf{x}}_e^1\end{bmatrix}\right)'\right]+\left(\begin{bmatrix}\hat{\mathbf{x}}_c^2\\ \hat{\mathbf{x}}_e^1\end{bmatrix}-\begin{bmatrix}\hat{\mathbf{x}}_c^{01}\\ \hat{\mathbf{x}}_{Me}^{01}\end{bmatrix}\right)\left(\begin{bmatrix}\hat{\mathbf{x}}_c^2\\ \hat{\mathbf{x}}_e^1\end{bmatrix}-\begin{bmatrix}\hat{\mathbf{x}}_c^{01}\\ \hat{\mathbf{x}}_{Me}^{01}\end{bmatrix}\right)'\right\}\mu_{2|1} \\
&\triangleq \left\{\begin{bmatrix} P_c^1 & P_{ce}^1 \\ P_{ec}^1 & P_e^1 \end{bmatrix}+\begin{bmatrix} \tilde{P}_c^1 & \tilde{P}_{Mce}^1 \\ \tilde{P}_{Mec}^1 & \tilde{P}_{Me}^1 \end{bmatrix}\right\}\mu_{1|1}+\left\{\begin{bmatrix} P_c^2 & P_{ce}^{21} \\ P_{ec}^{12} & P_e^1 \end{bmatrix}+\begin{bmatrix} \tilde{P}_c^2 & 0 \\ 0 & 0 \end{bmatrix}\right\}\mu_{2|1} \qquad \text{(A.0.1)}
\end{aligned}
$$

where $P_{ce}^{21}=0$ and $P_{ec}^{12}=0$, since the estimates of different components from different modes are assumed to be uncorrelated. Then (1.2.12) is obtained.

Appendix B

Unbiased Mixing for BLUE

Let us consider a generalized form for the unbiased mixing based on the best linear unbiased estimator (BLUE). Assuming a "base" state vector as

$$\mathbf{x} \triangleq \begin{bmatrix} \mathbf{x}_e^1 \\ \mathbf{x}_c \\ \mathbf{x}_e^2 \end{bmatrix} \tag{B.0.1}$$

and the state vectors of the IMM mode 1 and mode 2 as, respectively,

$$\mathbf{x}^1 = \begin{bmatrix} G_e^1 G_a^1 \mathbf{0} \\ G_b^1 G_c^1 \mathbf{0} \\ \mathbf{0} \;\; \mathbf{0} \;\; \mathbf{0} \end{bmatrix} \mathbf{x} = \begin{bmatrix} G_e^1 \mathbf{x}_e^1 + G_a^1 \mathbf{x}_c \\ G_b^1 \mathbf{x}_e^1 + G_c^1 \mathbf{x}_c \\ \mathbf{0} \end{bmatrix} \tag{B.0.2}$$

$$\mathbf{x}^2 = \begin{bmatrix} \mathbf{0} \;\; \mathbf{0} \;\; \mathbf{0} \\ \mathbf{0} G_c^2 G_b^2 \\ \mathbf{0} G_a^2 G_e^2 \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{0} \\ G_b^2 \mathbf{x}_e^2 + G_c^2 \mathbf{x}_c \\ G_e^2 \mathbf{x}_e^2 + G_a^2 \mathbf{x}_c \end{bmatrix} \tag{B.0.3}$$

where, without loss of generality, the diagonal non-zero matrices G_e^1, G_e^2 and $G_c^i, i = 1, 2$ are $m \times m$, $n \times n$ and $p \times p$ invertible matrices, respectively. The invertibility of the matrices guarantee the 1-on-1 mapping between the local state vector and the corresponding components of the "base" state vector. Then

the state of mode 1 is a linear combination of certain components of the state of interest $\mathbf{x}$; the state of mode 2 is a linear combination of certain another components of the state of interest $\mathbf{x}$. At the same time, each mode may have certain extra components the other does not have ($\mathbf{x}_e^1$ or $\mathbf{x}_e^2$) and also shares certain common components both have ($\mathbf{x}_c$).

The standard mixing of the estimates for the mode 1 filter is

$$\hat{\mathbf{x}}^{01} = \hat{\mathbf{x}}^1 \mu_{1|1} + \hat{\mathbf{x}}^2 \mu_{2|1} \tag{B.0.4}$$

$$= \begin{bmatrix} G_e^1 G_a^1 \mathbf{0} \\ G_b^1 G_c^1 \mathbf{0} \\ \mathbf{0} \quad \mathbf{0} \, \mathbf{0} \end{bmatrix} \hat{\mathbf{x}} \mu_{1|1} + \begin{bmatrix} \mathbf{0} \, \mathbf{0} \quad \mathbf{0} \\ \mathbf{0} G_c^2 G_b^2 \\ \mathbf{0} G_a^2 G_e^2 \end{bmatrix} \hat{\mathbf{x}} \mu_{2|1} \tag{B.0.5}$$

$$= \begin{bmatrix} (G_e^1 \hat{\mathbf{x}}_e^1 + G_a^1 \hat{\mathbf{x}}_c) \mu_{1|1} \\ (G_b^1 \hat{\mathbf{x}}_e^1 + G_c^1 \hat{\mathbf{x}}_c) \mu_{1|1} + (G_b^2 \hat{\mathbf{x}}_e^2 + G_c^2 \hat{\mathbf{x}}_c) \mu_{2|1} \\ (G_e^2 \hat{\mathbf{x}}_e^2 + G_a^2 \hat{\mathbf{x}}_c) \mu_{2|1} \end{bmatrix} \tag{B.0.6}$$

$$\triangleq \begin{bmatrix} \hat{\mathbf{x}}_e^{01} \\ \hat{\mathbf{x}}_c^{01} \\ \hat{\mathbf{x}}_*^{01} \end{bmatrix} \tag{B.0.7}$$

and for the mode 2 filter it is

$$\hat{\mathbf{x}}^{02} = \hat{\mathbf{x}}^1 \mu_{1|2} + \hat{\mathbf{x}}^2 \mu_{2|2} \tag{B.0.8}$$

$$= \begin{bmatrix} G_e^1 G_a^1 \mathbf{0} \\ G_b^1 G_c^1 \mathbf{0} \\ \mathbf{0} \quad \mathbf{0} \, \mathbf{0} \end{bmatrix} \hat{\mathbf{x}} \mu_{1|2} + \begin{bmatrix} \mathbf{0} \, \mathbf{0} \quad \mathbf{0} \\ \mathbf{0} G_c^2 G_b^2 \\ \mathbf{0} G_a^2 G_e^2 \end{bmatrix} \hat{\mathbf{x}} \mu_{2|2} \tag{B.0.9}$$

$$= \begin{bmatrix} (G_e^1 \hat{\mathbf{x}}_e^1 + G_a^1 \hat{\mathbf{x}}_c) \mu_{1|2} \\ (G_b^1 \hat{\mathbf{x}}_e^1 + G_c^1 \hat{\mathbf{x}}_c) \mu_{1|2} + (G_b^2 \hat{\mathbf{x}}_e^2 + G_c^2 \hat{\mathbf{x}}_c) \mu_{2|2} \\ G_e^2 \hat{\mathbf{x}}_e^2 + (G_a^2 \hat{\mathbf{x}}_c) \mu_{2|2} \end{bmatrix} \tag{B.0.10}$$

$$\triangleq \begin{bmatrix} \hat{\mathbf{x}}_*^{02} \\ \hat{\mathbf{x}}_c^{02} \\ \hat{\mathbf{x}}_e^{02} \end{bmatrix} \tag{B.0.11}$$

Note that the last n components of (B.0.6) are ignored by the mode 1 filter while the last n components of (B.0.10) are multiplied by a factor $\mu_{2|2}$ less than one; the first m components of (B.0.10) are ignored by the mode 2 filter while the first m components of (B.0.6) are multiplied by a factor $\mu_{1|1}$ less than one. This would lead to estimation bias for the components $\mathbf{x}_e^1$ and $\mathbf{x}_e^2$ in each IMM cycle.

The remedy is replacing estimates of mode 1 and mode 2 as follows

$$\hat{\mathbf{x}}_M^1 = \begin{bmatrix} G_e^1 \hat{\mathbf{x}}_e^1 + G_a^1 \hat{\mathbf{x}}_c \\ G_b^1 \hat{\mathbf{x}}_e^1 + G_c^1 \hat{\mathbf{x}}_c \\ G_e^2 \hat{\mathbf{x}}_e^2 + G_a^2 \hat{\mathbf{x}}_c \end{bmatrix} \tag{B.0.12}$$

and

$$\hat{\mathbf{x}}_M^2 = \begin{bmatrix} G_e^1\hat{\mathbf{x}}_e^1 + G_a^1\hat{\mathbf{x}}_c \\ G_b^2\hat{\mathbf{x}}_e^2 + G_c^2\hat{\mathbf{x}}_c \\ G_e^2\hat{\mathbf{x}}_e^2 + G_a^2\hat{\mathbf{x}}_c \end{bmatrix} \tag{B.0.13}$$

Then the modified estimate mixing for the mode 1 filter and the mode 2 filter are

$$\hat{\mathbf{x}}_M^{01} = \hat{\mathbf{x}}_M^1\mu_{1|1} + \hat{\mathbf{x}}_M^2\mu_{2|1} \tag{B.0.14}$$

$$= \begin{bmatrix} (G_e^1\hat{\mathbf{x}}_e^1 + G_a^1\hat{\mathbf{x}}_c)\mu_{1|1} + (G_e^1\hat{\mathbf{x}}_e^1 + G_a^1\hat{\mathbf{x}}_c)\mu_{2|1} \\ (G_b^1\hat{\mathbf{x}}_e^1 + G_c^1\hat{\mathbf{x}}_c)\mu_{1|1} + (G_b^2\hat{\mathbf{x}}_e^2 + G_c^2\hat{\mathbf{x}}_c)\mu_{2|1} \\ (G_e^2\hat{\mathbf{x}}_e^2 + G_a^2\hat{\mathbf{x}}_c)\mu_{1|1} + (G_e^2\hat{\mathbf{x}}_e^2 + G_a^2\hat{\mathbf{x}}_c)\mu_{2|1} \end{bmatrix} \tag{B.0.15}$$

$$= \begin{bmatrix} G_e^1\hat{\mathbf{x}}_e^1 + G_a^1\hat{\mathbf{x}}_c \\ (G_b^1\hat{\mathbf{x}}_e^1 + G_c^1\hat{\mathbf{x}}_c)\mu_{1|1} + (G_b^2\hat{\mathbf{x}}_e^2 + G_c^2\hat{\mathbf{x}}_c)\mu_{2|1} \\ G_e^2\hat{\mathbf{x}}_e^2 + G_a^2\hat{\mathbf{x}}_c \end{bmatrix} \tag{B.0.16}$$

and

$$\hat{\mathbf{x}}_M^{02} = \hat{\mathbf{x}}_M^1 \mu_{1|2} + \hat{\mathbf{x}}_M^2 \mu_{2|2} \tag{B.0.17}$$

$$= \begin{bmatrix} (G_e^1 \hat{\mathbf{x}}_e^1 + G_a^1 \hat{\mathbf{x}}_c)\mu_{1|2} + (G_e^1 \hat{\mathbf{x}}_e^1 + G_a^1 \hat{\mathbf{x}}_c)\mu_{2|2} \\ (G_b^1 \hat{\mathbf{x}}_e^1 + G_c^1 \hat{\mathbf{x}}_c)\mu_{1|2} + (G_b^2 \hat{\mathbf{x}}_e^2 + G_c^2 \hat{\mathbf{x}}_c)\mu_{2|2} \\ (G_e^2 \hat{\mathbf{x}}_e^2 + G_a^2 \hat{\mathbf{x}}_c)\mu_{1|2} + (G_e^2 \hat{\mathbf{x}}_e^2 + G_a^2 \hat{\mathbf{x}}_c)\mu_{2|2} \end{bmatrix} \tag{B.0.18}$$

$$= \begin{bmatrix} G_e^1 \hat{\mathbf{x}}_e^1 + G_a^1 \hat{\mathbf{x}}_c \\ (G_b^1 \hat{\mathbf{x}}_e^1 + G_c^1 \hat{\mathbf{x}}_c)\mu_{1|2} + (G_b^2 \hat{\mathbf{x}}_e^2 + G_c^2 \hat{\mathbf{x}}_c)\mu_{2|2} \\ G_e^2 \hat{\mathbf{x}}_e^2 + G_a^2 \hat{\mathbf{x}}_c \end{bmatrix} \tag{B.0.19}$$

Given the augmented covariance matrices of (B.0.2) and (B.0.3) as

$$P^1 \triangleq \begin{bmatrix} P_e^1 & P_{ec}^1 & \mathbf{0} \\ P_{ce}^1 & P_c^1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{B.0.20}$$

and

$$P^2 \triangleq \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & P_c^2 & P_{ce}^2 \\ \mathbf{0} & P_{ec}^2 & P_e^2 \end{bmatrix} \tag{B.0.21}$$

the corresponding covariance matrices for the unbiased mixing can be obtained accordingly as follows.

For mode 1 filter, the corresponding augmented covariance matrix is

$$P_M^{01} = \left\{ \begin{bmatrix} P_e^1 & P_{ec}^1 & * \\ P_{ce}^1 & P_c^1 & * \\ * & * & * \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & * \\ \mathbf{0} & \tilde{P}_c^{1,01} & * \\ * & * & * \end{bmatrix} \right\} \mu_{1|1} + \left\{ \begin{bmatrix} P_e^1 & \mathbf{0} & * \\ \mathbf{0} & P_c^2 & * \\ * & * & * \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & * \\ \mathbf{0} & \tilde{P}_c^{2,01} & * \\ * & * & * \end{bmatrix} \right\} \mu_{2|1} \tag{B.0.22}$$

where "*" implies the elements ignored by mode 1 filter no matter what they are; the spread-of-the-means terms are

$$\tilde{P}_c^{1,01} = (\hat{\mathbf{x}}_c^1 - \hat{\mathbf{x}}_c^{01})(\hat{\mathbf{x}}_c^1 - \hat{\mathbf{x}}_c^{01})' \tag{B.0.23}$$

$$\tilde{P}_c^{2,01} = (\hat{\mathbf{x}}_c^2 - \hat{\mathbf{x}}_c^{01})(\hat{\mathbf{x}}_c^2 - \hat{\mathbf{x}}_c^{01})' \tag{B.0.24}$$

Similarly, for mode 2 filter, the corresponding augmented covariance matrix is

$$P_M^{02} = \left\{ \begin{bmatrix} * & * & * \\ * & P_c^1 & \mathbf{0} \\ * & \mathbf{0} & P_e^2 \end{bmatrix} + \begin{bmatrix} * & * & * \\ * & \tilde{P}_c^{1,02} & \mathbf{0} \\ * & \mathbf{0} & \mathbf{0} \end{bmatrix} \right\} \mu_{1|2} + \left\{ \begin{bmatrix} * & * & * \\ * & P_c^2 & P_{ce}^2 \\ * & P_{ec}^2 & P_e^2 \end{bmatrix} + \begin{bmatrix} * & * & * \\ * & \tilde{P}_c^{2,02} & \mathbf{0} \\ * & \mathbf{0} & \mathbf{0} \end{bmatrix} \right\} \mu_{2|2} \tag{B.0.25}$$

where "*" implies the elements ignored by mode 2 filter no matter what they

are; the spread-of-the-means terms are

$$\tilde{P}_c^{1,02}=(\hat{\mathbf{x}}_c^1-\hat{\mathbf{x}}_c^{02})(\hat{\mathbf{x}}_c^1-\hat{\mathbf{x}}_c^{02})' \tag{B.0.26}$$

$$\tilde{P}_c^{2,02}=(\hat{\mathbf{x}}_c^2-\hat{\mathbf{x}}_c^{02})(\hat{\mathbf{x}}_c^2-\hat{\mathbf{x}}_c^{02})' \tag{B.0.27}$$

In most situations, we are dealing with the following simple case (with $\mathbf{x}$ as the state of interest):

$$\mathbf{x}^1=\begin{bmatrix} I_m & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}\mathbf{x}=\begin{bmatrix} \mathbf{x}_e^1 \\ \mathbf{x}_c \\ \mathbf{0} \end{bmatrix} \tag{B.0.28}$$

$$\mathbf{x}^2=\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_n \end{bmatrix}\mathbf{x}=\begin{bmatrix} \mathbf{0} \\ \mathbf{x}_c \\ \mathbf{x}_e^2 \end{bmatrix} \tag{B.0.29}$$

where $I_r, r=m,n,p$, are the $r\times r$ identity matrices. Further setting I_n in (B.0.29) as all-zero matrix and eliminating the corresponding rows and columns, we get the special case discussed in Section II with the state of interest as $\mathbf{x}=\begin{bmatrix} \mathbf{x}_e^1 \\ \mathbf{x}_c \end{bmatrix}$.

Appendix C

Jacobian Matrix for Thrusting Mode

The Jacobian A for the TM, under the assumption that α (under subsonic speed) and τ are constant, can be shown as (the time arguments are omitted)

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} & a_{48} \\ 0 & 0 & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & a_{58} \\ 0 & 0 & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} & a_{68} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{C.0.1}$$

where, note that D depends on both altitude z and velocity v,

$$a_{43} = \alpha\,\alpha_m \frac{\partial D}{\partial z}\dot{x} \tag{C.0.2}$$

$$a_{44} = \tau \frac{V^2 - \dot{x}^2}{V^3} + \alpha\,\alpha_m D + \alpha\,\alpha_m \dot{x}\frac{\partial D}{\partial \dot{x}} \tag{C.0.3}$$

$$a_{45} = \tau\dot{x}\frac{-\dot{y}}{V^3} + \alpha\,\alpha_m \dot{x}\frac{\partial D}{\partial \dot{y}} \tag{C.0.4}$$

$$a_{46} = \tau\dot{x}\frac{-\dot{z}}{V^3} + \alpha\,\alpha_m \dot{x}\frac{\partial D}{\partial \dot{z}} \tag{C.0.5}$$

$$a_{47} = D\dot{x} \tag{C.0.6}$$

$$a_{48} = \frac{\dot{x}}{V} \tag{C.0.7}$$

$$a_{53} = \alpha\,\alpha_m \frac{\partial D}{\partial z}\dot{y} \tag{C.0.8}$$

$$a_{54} = \tau\dot{y}\frac{-\dot{x}}{V^3} + \alpha\,\alpha_m \dot{y}\frac{\partial D}{\partial \dot{x}} \tag{C.0.9}$$

$$a_{55} = \tau \frac{V^2 - \dot{y}^2}{V^3} + \alpha\,\alpha_m D + \alpha\,\alpha_m \dot{y}\frac{\partial D}{\partial \dot{y}} \tag{C.0.10}$$

$$a_{56} = \tau\dot{y}\frac{-\dot{z}}{V^3} + \alpha\,\alpha_m \dot{y}\frac{\partial D}{\partial \dot{z}} \tag{C.0.11}$$

$$a_{57} = D\dot{y} \tag{C.0.12}$$

$$a_{58} = \frac{\dot{y}}{V} \tag{C.0.13}$$

$$a_{63} = \alpha\,\alpha_m \frac{\partial D}{\partial z}\dot{z} \tag{C.0.14}$$

$$a_{64} = \tau\dot{z}\frac{-\dot{x}}{V^3} + \alpha\,\alpha_m \dot{z}\frac{\partial D}{\partial \dot{x}} \tag{C.0.15}$$

$$a_{65} = \tau\dot{z}\frac{-\dot{y}}{V^3} + \alpha\,\alpha_m \dot{z}\frac{\partial D}{\partial \dot{y}} \tag{C.0.16}$$

$$a_{66} = \tau \frac{V^2 - \dot{z}^2}{V^3} + \alpha\,\alpha_m D + \alpha\,\alpha_m \dot{z}\frac{\partial D}{\partial \dot{z}} \tag{C.0.17}$$

$$a_{67} = D\dot{z} \tag{C.0.18}$$

$$a_{68} = \frac{\dot{z}}{V} \tag{C.0.19}$$

with

$$\frac{\partial D}{\partial z} = \frac{c\rho(z)V}{2} = -cD \tag{C.0.20}$$

$$\frac{\partial D}{\partial \dot{x}} = -\frac{\rho(z)\dot{x}}{2V} = D\frac{\dot{x}}{V^2} \tag{C.0.21}$$

$$\frac{\partial D}{\partial \dot{y}} = -\frac{\rho(z)\dot{y}}{2V} = D\frac{\dot{y}}{V^2} \tag{C.0.22}$$

$$\frac{\partial D}{\partial \dot{z}} = -\frac{\rho(z)\dot{z}}{2V} = D\frac{\dot{z}}{V^2} \tag{C.0.23}$$

The Jacobian for the ballistic mode can be easily obtained by setting τ to zero in above expressions and eliminating the last row and column.

Appendix D

Taylor Expansion Approximation for LMMSE Fuser

By the first order Taylor expansion, we have

$$g(\mathbf{x}^i) \approx g(\hat{\mathbf{x}}^i) + G^i(\mathbf{x}^i - \hat{\mathbf{x}}^i) \tag{D.0.1}$$

where G^i is the Jacobian of $g(\mathbf{x}^i)$ evaluated at $\hat{\mathbf{x}}^i$, as defined in (4.3.5). Then (with the knowledge that $\mathbf{x}^j = g(\mathbf{x}^i)$)

$$\begin{aligned} \hat{\mathbf{x}}^j - g(\hat{\mathbf{x}}^i) &= \hat{\mathbf{x}}^j - g(\mathbf{x}^i) + G^i(\mathbf{x}^i - \hat{\mathbf{x}}^i) \\ &= G^i(\mathbf{x}^i - \hat{\mathbf{x}}^i) - (\mathbf{x}^j - \hat{\mathbf{x}}^j) \end{aligned} \tag{D.0.2}$$

So

$$\begin{aligned} P_{\mathbf{xz}} &\triangleq E\left[\left(\mathbf{x}^i - \hat{\mathbf{x}}^i\right)\left(\hat{\mathbf{x}}^j - g(\hat{\mathbf{x}}^i)\right)'\right] \\ &\approx E\left[\left(\mathbf{x}^i - \hat{\mathbf{x}}^i\right)\left(G^i(\mathbf{x}^i - \hat{\mathbf{x}}^i) - (\mathbf{x}^j - \hat{\mathbf{x}}^j)\right)'\right] \\ &= E\left[\left(\mathbf{x}^i - \hat{\mathbf{x}}^i\right)(\mathbf{x}^i - \hat{\mathbf{x}}^i)'\right](G^i)' - E\left[\left(\mathbf{x}^i - \hat{\mathbf{x}}^i\right)(\mathbf{x}^j - \hat{\mathbf{x}}^j)'\right] \\ &= P^i(G^i)' - P^{ij} \end{aligned} \tag{D.0.3}$$

and

$$
\begin{aligned}
P_{\mathbf{zz}} \triangleq & E\left[\left(\hat{\mathbf{x}}^j - g(\hat{\mathbf{x}}^i)\right)\left(\hat{\mathbf{x}}^j - g(\hat{\mathbf{x}}^i)\right)'\right] \\
\approx & E\left[\left(G^i(\mathbf{x}^i - \hat{\mathbf{x}}^i) - (\mathbf{x}^j - \hat{\mathbf{x}}^j)\right)\left(G^i(\mathbf{x}^i - \hat{\mathbf{x}}^i) - (\mathbf{x}^j - \hat{\mathbf{x}}^j)\right)'\right] \\
= & P^j - G^i P^{ij} - P^{ji}(G^i)' + G^i P^i (G^i)' \qquad \text{(D.0.4)}
\end{aligned}
$$

Appendix E

Target State-Dependent Process Noise Covariance Matrix for NCT Model

As shown in [16], the target state-dependent process noise (in the Cartesian velocity model) is

$$\mathbf{v}_a^i[\mathbf{x}_a^i(k)] = D_a^i[\mathbf{x}_a^i(k)]\mathbf{v}_c^i(k) \tag{E.0.1}$$

where $\mathbf{v}_c^i(k) \sim \mathcal{N}(0, I_4)$ and

$$D_a^i[\mathbf{x}_a^i(k)] = S_a^i[\mathbf{x}_a^i(k)]V_a^i(k) \tag{E.0.2}$$

with

$$S_a^i[\mathbf{x}_a^i(k)]=\begin{bmatrix} \sqrt{q_a}s_1(k) & 0 & 0 & 0 \\ 0 & \sqrt{q_a}s_3(k)-\sqrt{q_\Omega}\dot{y}(k) & 0 \\ \sqrt{q_\Omega}s_2(k) & 0 & 0 & 0 \\ 0 & \sqrt{q_a}s_4(k) & \sqrt{q_\Omega}\dot{x}(k) & 0 \\ 0 & 0 & 0 & \sqrt{q_\Omega} \end{bmatrix} \tag{E.0.3}$$

$$V_a^i(k)=\begin{bmatrix} \sqrt{T_a^3/3} & 0 & 0 & 0 \\ \frac{\sqrt{3T_a}}{2} & \frac{\sqrt{T_a}}{2} & 0 & 0 \\ 0 & 0 & \sqrt{T_a^3/3} & 0 \\ 0 & 0 & \frac{\sqrt{3T_a}}{2} & \frac{\sqrt{T_a}}{2} \end{bmatrix} \tag{E.0.4}$$

where q_a and q_Ω are the continuous-time process noise PSDs and

$$s_1(k)=\frac{\dot{x}(k)}{\sqrt{\dot{x}^2(k)+\dot{y}^2(k)}} \tag{E.0.5}$$

$$s_2(k)=\frac{\dot{y}(k)}{\sqrt{\dot{x}^2(k)+\dot{y}^2(k)}} \tag{E.0.6}$$

$$s_3(k)=\frac{\dot{x}(k)-T_a\Omega(k)\dot{y}(k)}{\sqrt{\dot{x}^2(k)+\dot{y}^2(k)}} \tag{E.0.7}$$

$$s_4(k)=\frac{\dot{y}(k)+T_a\Omega(k)\dot{x}(k)}{\sqrt{\dot{x}^2(k)+\dot{y}^2(k)}} \tag{E.0.8}$$

We have the corresponding target state-dependent process noise covariance

matrix (the time index k is ignored for conciseness)

$$Q_a^i[\mathbf{x}_a^i(k)] \triangleq E\left\{\mathbf{v}_a^i[\mathbf{x}_a^i(k)]\mathbf{v}_a^i[\mathbf{x}_a^i(k)]'\right\} \tag{E.0.9}$$

$$= E\left\{\left[D_a^i[\mathbf{x}_a^i(k)]\mathbf{v}_c^i(k)\right]\left[D_a^i[\mathbf{x}_a^i(k)]\mathbf{v}_c^i(k)\right]'\right\} \tag{E.0.10}$$

$$= \begin{bmatrix} \frac{T_a^3}{3}s_1^2q_a & \frac{T_a^2}{2}s_1s_3q_a & \frac{T_a^3}{3}s_1s_2q_a & \frac{T_a^2}{2}s_1s_4q_a & 0 \\ \frac{T_a^2}{2}s_1s_3q_a & \frac{T_a^3}{3}\dot{y}^2q_\Omega + T_as_3^2q_a & \frac{T_a^2}{2}s_2s_3q_a & -\frac{T_a^3}{3}\dot{x}\dot{y}q_\Omega + T_as_3s_4q_a & -\frac{T_a^2}{2}\dot{y}q_\Omega \\ \frac{T_a^3}{3}s_1s_2q_a & \frac{T_a^2}{2}s_2s_3q_a & \frac{T_a^3}{3}s_2^2q_a & \frac{T_a^2}{2}s_2s_4q_a & 0 \\ \frac{T_a^2}{2}s_1s_4q_a & -\frac{T_a^3}{3}\dot{x}\dot{y}q_\Omega + T_as_3s_4q_a & \frac{T_a^2}{2}s_2s_4q_a & \frac{T_a^3}{3}\dot{x}^2q_\Omega + T_as_4^2q_a & \frac{T_a^2}{2}\dot{x}q_\Omega \\ 0 & -\frac{T_a^2}{2}\dot{y}q_\Omega & 0 & \frac{T_a^2}{2}\dot{x}q_\Omega & T_aq_\Omega \end{bmatrix} \tag{E.0.11}$$

Guideline of Process Noise Intensity Selection for the NCT Model

Observing the diagonal elements of $Q_a^i[\mathbf{x}_a^i(k)]$ in (E.0.11), the process noise induced RMS change rate in the velocity (x- and y- coordinate) and in the turn rate over a sampling interval T_a are

$$d_a^x \triangleq \frac{\sqrt{\frac{T_a^3}{3}\dot{y}^2q_\Omega + T_as_3^2q_a}}{T_a} \tag{E.0.12}$$

$$d_a^y \triangleq \frac{\sqrt{\frac{T_a^3}{3}\dot{x}^2q_\Omega + T_as_4^2q_a}}{T_a} \tag{E.0.13}$$

$$d_\Omega \triangleq \frac{\sqrt{T_aq_\Omega}}{T_a} \tag{E.0.14}$$

respectively. It is obvious that the choices of the design values d_a (d_a^x and d_a^y) and d_Ω need to be considered simultaneously.

To make things simpler, we consider an extreme case with $\dot{x} = 0$ (similar case

for $\dot{y} = 0$). Given a target with speed V_s (which is equal to the magnitude of $\dot{y}$ in this extreme case), we have

$$s_1^* = 0 \tag{E.0.15}$$

$$s_2^* = 1 \tag{E.0.16}$$

$$s_3^* = -T_a\Omega(k) \tag{E.0.17}$$

$$s_4^* = 1 \tag{E.0.18}$$

this leads to

$$d_a^{x*} = \frac{\sqrt{\frac{T_a^3}{3}V_s{}^2 q_\Omega + T_a^3\Omega^2(k)q_a}}{T_a} \tag{E.0.19}$$

$$d_a^{y*} = \frac{\sqrt{T_a q_a}}{T_a} \tag{E.0.20}$$

Since the assumption is $\dot{x} = 0$, the process noise induced RMS change in X-coordinate velocity should be more sensitive compared with that in Y-coordinate. Given a conjectured choice as d_Ω and d_a, the process noise induced RMS change in X-coordinate velocity contributed by the PSD q_Ω and q_a are, respectively,

$$\begin{aligned} d_a^{x*}(q_\Omega, q_a = 0) &= \frac{\sqrt{\frac{T_a^3}{3}V_s{}^2 q_\Omega}}{T_a} \\ &= \frac{T_a \dot{y} d_\Omega^*}{\sqrt{3}} \end{aligned} \tag{E.0.21}$$

$$\begin{aligned} d_a^{x*}(q_\Omega = 0, q_a) &= \frac{\sqrt{T_a^3\Omega^2(k)q_a}}{T_a} \\ &= \sqrt{T_a\Omega^2(k)q_a} \end{aligned} \tag{E.0.22}$$

It is important to compare those two values. The process noise induced RMS change rate for both the linear velocity and the turn rate should be within reasonable ranges over time interval T_a.

Remarks

Without loss of generality, based on (E.0.20), (E.0.21) and (E.0.22), the selections of the design values d_a and d_Ω for the NCT model are shown as (sequentially)

(i) Select d_Ω, then $d_a^{x*}(q_\Omega, q_a = 0)$ has been obtained and should be in a reasonable range.

(ii) Select d_a, then $d_a^{x*}(q_\Omega = 0, q_a)$ has been obtained and should be in a reasonable range. Further, d_a^{x*} should be not too big and d_a^{y*} should be not too small.

Appendix F

MC Results for Sample Crosscorrelation

The *sample crosscorrelation coefficient* between the lth component of $\mathbf{x}^i$ and the hth component of $\mathbf{x}^j$ in M MC runs at a particular point in time (not indicated, for conciseness) is

$$\hat{\rho}^M_{\mathbf{x}^i_l \mathbf{x}^j_h} \triangleq \frac{\sum_{m=1}^{M} (\hat{\mathbf{x}}^i_{l,m} - \mathbf{x}^i_l)(\hat{\mathbf{x}}^j_{h,m} - \mathbf{x}^i_h)}{\sqrt{\left[\sum_{m=1}^{M} (\hat{\mathbf{x}}^i_{l,m} - \mathbf{x}^i_l)^2\right]\left[\sum_{m=1}^{M} (\hat{\mathbf{x}}^j_{h,m} - \mathbf{x}^j_h)^2\right]}} \tag{F.0.1}$$

The sample crosscorrelation coefficients of different heterogeneous components from 1000 MC runs, for the scenario described in Section 4.5, are shown in Figs. F.0.1–F.0.4. It can be seen that the "common process noise effect", driven by real maneuvers here, leads to significant crosscorrelation between the estimation errors from the heterogeneous local sensors. Furthermore, both positive and negative crosscorrelation are observed. This motivates the geometry-based "functional model" discussed in Appendix G.

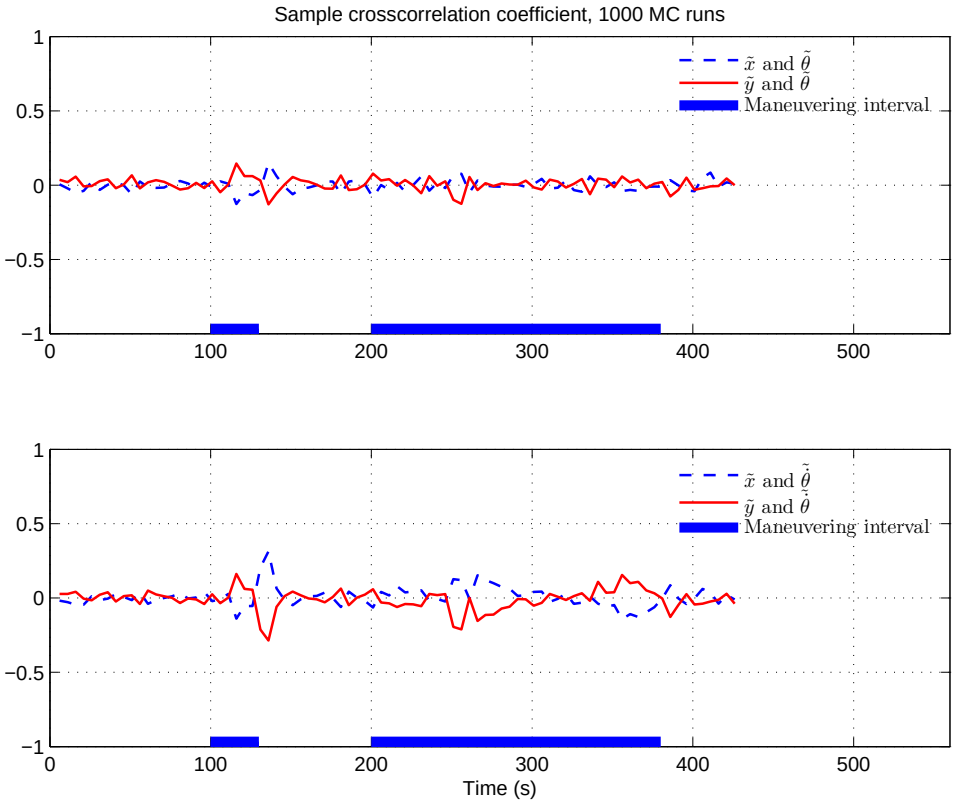


Figure F.0.1: Sample crosscorrelation for $\tilde{x}$ and $\tilde{y}$ with $\tilde{\theta}$ and $\tilde{\tilde{\theta}}$.

Appendix G

An Approximation Technique for Crosscovariance Matrix

By considering the steady-state case for a KF, an approximate technique for the evaluation of the crosscovariance matrix in the homogeneous case has been developed recently [7]. This technique, which relies on the crosscorrelation coefficients between the local estimates of the same state components, namely,

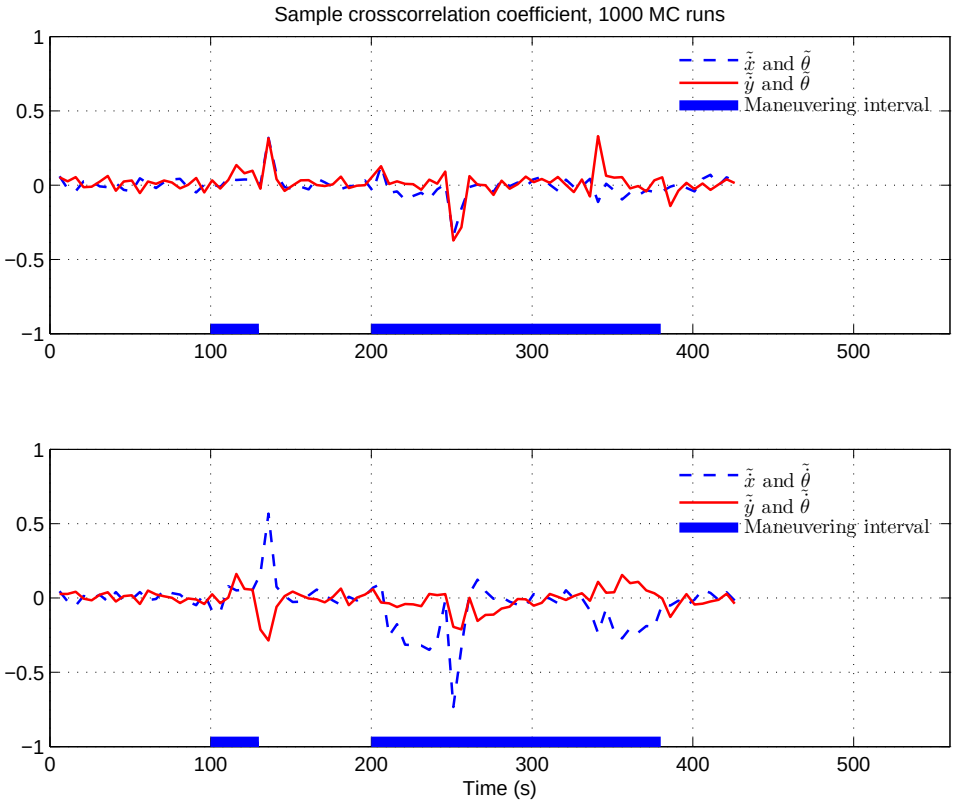


Figure F.0.2: Sample crosscorrelation for $\tilde{x}$ and $\tilde{y}$ with $\tilde{\theta}$ and $\tilde{\dot{\theta}}$.

position and velocity and the maneuvering indices at the different sensors, can be extended to heterogeneous case as follows.

The components of the state $\mathbf{x}^i$ are grouped by coordinates as follows

$$\mathbf{x}_1^i = [\, x \quad \dot{x} \,]' \tag{G.0.1}$$

$$\mathbf{x}_2^i = [\, y \quad \dot{y} \,]' \tag{G.0.2}$$

which can be "aligned" with those of $\mathbf{x}^j$ as it will be shown in the sequel. Then the first components of $\mathbf{x}_c^i$, $c = 1, 2$ and $\mathbf{x}^j$ are position and the second components are velocity, albeit in different spaces. The covariance submatrix

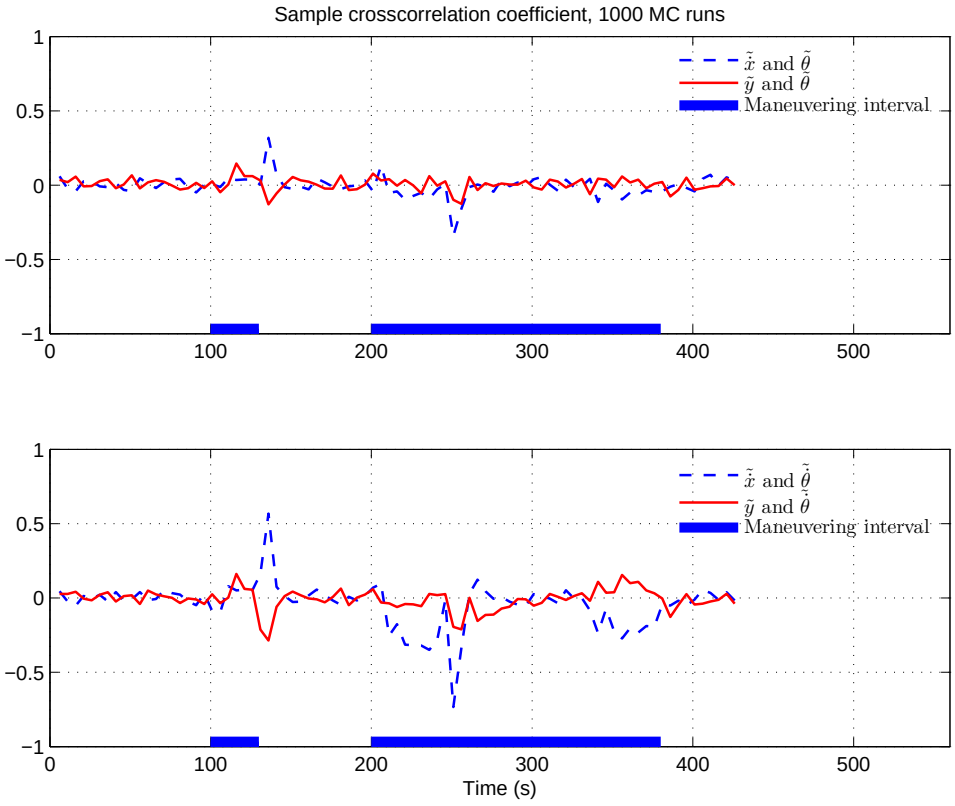


Figure F.0.3: Sample crosscorrelation for $\tilde{x}$ and $\tilde{y}$ with $\tilde{\theta}$ and $\dot{\tilde{\theta}}$.

P_c^i, corresponding to estimate $\hat{\mathbf{x}}_c^i$, follows from the corresponding elements of P^i.

Extending the crosscovariance matrix approximation technique for the homogeneous case in [7] to the heterogeneous case, we can then approximately reconstruct the crosscovariance matrix elements using the following expression

$$P_{c,lh}^{ij} = a_{c,lh}^{ij}\,\rho_{c,lh}^{ij}\,\sqrt{P_c^i(l,l)P^j(h,h)} \qquad c = 1,2;\; l,h = 1,2 \tag{G.0.3}$$

where $c = 1,2$ correspond to the first and the second part of the state $\mathbf{x}^i$; $l,h = 1$ represent position and $l,h = 2$ represent velocity; $\rho_{c,lh}^{ij}$ is the *maximum cross-*

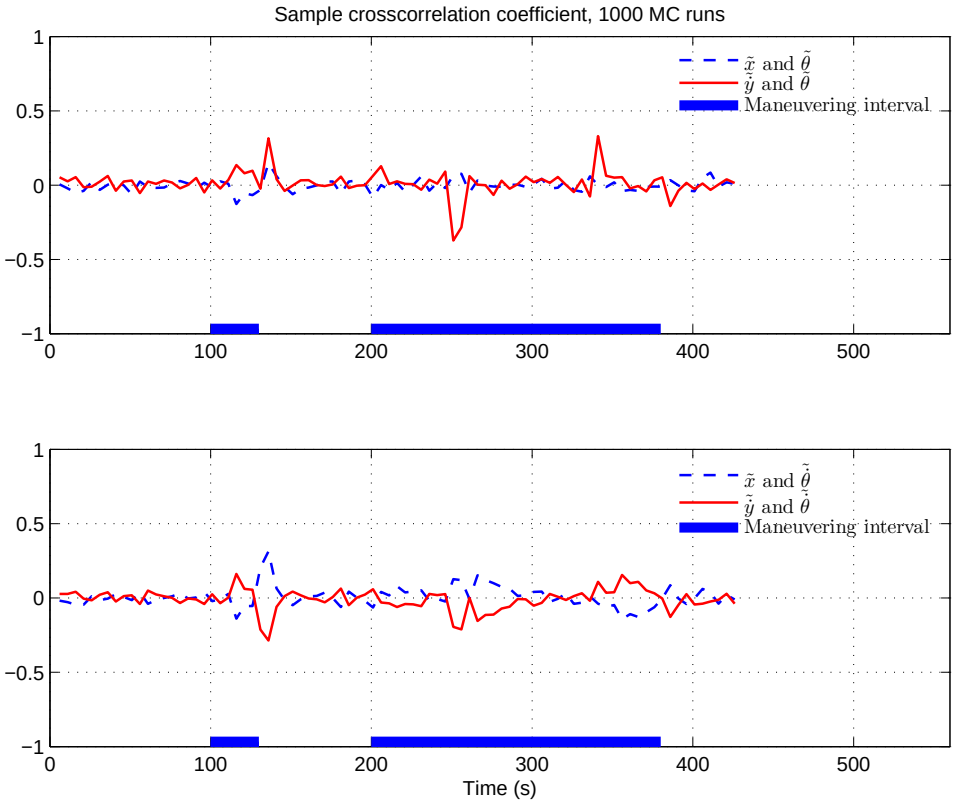


Figure F.0.4: Sample crosscorrelation for $\tilde{x}$ and $\tilde{y}$ with $\tilde{\theta}$ and $\tilde{\theta}$.

correlation coefficient of the estimation errors in $\mathbf{x}_{c,l}^{i}$ and $\mathbf{x}_{h}^{j}$ and $a_{c,lh}^{ij}$ is geometry-dependent *adjustment factor* in the crosscorrelation coefficient. This factor is discussed below.

The maximum crosscorrelation coefficients are denoted as ρ_{pp} for position-position, ρ_{pv} for position-velocity, ρ_{vp} for velocity-position and ρ_{vv} for velocity-

velocity. Then we have

$$\rho^{ij}_{c,lh} = \begin{cases} \rho_{pp} & l,h = 1 \\ \rho_{pv} & l = 1,\ h = 2 \\ \rho_{vp} & l = 2,\ h = 1 \\ \rho_{vv} & l,h = 2 \end{cases} \tag{G.0.4}$$

which are chosen (similarly to [7]) as $\rho_{pp} = 0.1$, $\rho_{pv} = \rho_{vp} = 0.15$ and $\rho_{vv} = 0.45$ in our simulations.

There are two state variable pairs, say, x-θ and y-θ, for the position-position crosscorrelation coefficient. The best way to quantify the crosscorrelation difference (accounts for the geometry) for those state variable pairs is based on the function (4.2.7).

From the Cartesian to polar coordinate transformation (see, e.g., [5]), one has the following proportionalities

$$\frac{\partial x}{\partial \theta} \propto -\sin\theta \tag{G.0.5}$$

$$\frac{\partial y}{\partial \theta} \propto \cos\theta \tag{G.0.6}$$

Based on the above, we have the following adjustment factors for the crosscorrelation coefficients of the pairs x-θ and y-θ

$$a^{ij}_{1,11} = -\sin\theta \tag{G.0.7}$$

$$a^{ij}_{2,11} = \cos\theta \tag{G.0.8}$$

Then the crosscorrelation coefficients of the pairs x-θ and y-θ are obtained from combining the adjustment factor (G.0.7)–(G.0.8) with (G.0.4) as

$$\rho_{x\theta} = -\rho_{pp} \sin\theta \tag{G.0.9}$$

$$\rho_{y\theta} = \rho_{pp} \cos\theta \tag{G.0.10}$$

Similarly, we have the adjustment factors

$$a_{c,lh}^{ij} = \begin{cases} -\sin(\theta), \; c = 1 \\ \cos(\theta), \; c = 2 \end{cases} \tag{G.0.11}$$

and the other crosscorrelation coefficients are

$$\rho_{x\dot{\theta}} = -\rho_{pv} \sin\theta \tag{G.0.12}$$

$$\rho_{y\dot{\theta}} = \rho_{pv} \cos\theta \tag{G.0.13}$$

$$\rho_{\dot{x}\theta} = -\rho_{vp} \sin\theta \tag{G.0.14}$$

$$\rho_{\dot{y}\theta} = \rho_{vp} \cos\theta \tag{G.0.15}$$

$$\rho_{\dot{x}\dot{\theta}} = -\rho_{vv} \sin\theta \tag{G.0.16}$$

$$\rho_{\dot{y}\dot{\theta}} = \rho_{vv} \cos\theta \tag{G.0.17}$$

Finally, the reconstructed crosscovariance matrix is

$$P^{ij} = \begin{bmatrix} P_1^{ij} \\ P_2^{ij} \end{bmatrix} \tag{G.0.18}$$

with the elements of each block in (G.0.18) given by (G.0.3) with (G.0.5)–

(G.0.17).

For the scenario described in Section 4.5, the RMSE in position and velocity under both "uncorr" and "corr" assumptions at some fusion times for the LMMSE fuser are shown in Table G.0.19.

Table G.0.19: RMSE in position and velocity for LMMSE fuser

Position					
	100 s	110 s	130 s	150 s	255 s
"uncorr"	27.6 m	37.5 m	33.6 m	28.9 m	30.3 m
"corr"	26.6 m	36.4 m	33.5 m	29.1 m	29.5 m
Velocity					
	100 s	110 s	130 s	150 s	255 s
"uncorr"	9.4 m/s	15.3 m/s	10.7 m/s	5.1 m/s	17.4 m/s
"corr"	9.5 m/s	14.9 m/s	10.8 m/s	6.1 m/s	18.2 m/s

Based on these results, which show negligible performance differences, it seems preferable to follow the "uncorr" assumption.

Appendix H

Choice of Estimator for the Passive Sensor

The guideline for deciding whether to use an IMM estimator or a (single model) KF can be quantified in terms of the target maneuvering index, which is the ratio between the standard deviation (RMS values) of the motion uncertainty and the measurement uncertainty [3] [11]. Namely, if this index is below 0.5 then there is no point in using an IMM.

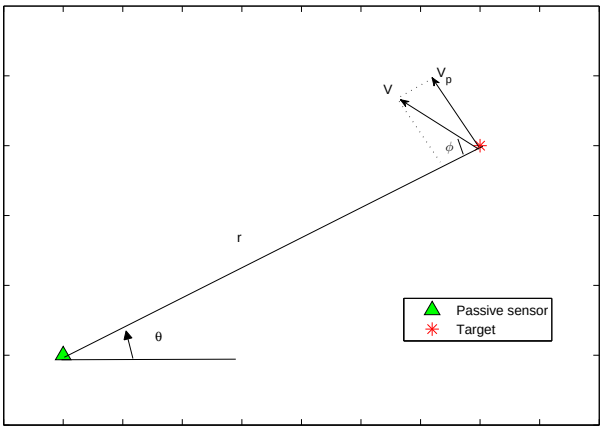


Figure H.0.5: Illustrated scenario for calculating target maneuvering index.

For the passive sensor considered, the maneuvering index can be calculated as follows. As shown in Fig. H.0.5, the angular velocity seen by the passive sensor

is

$$\dot{\theta}_p = \frac{V \sin\varphi}{r_p} \tag{H.0.1}$$

where V is the speed of the target and r_p is the range of the target from the passive sensor. Then the angular acceleration seen by the passive sensor is

$$\ddot{\theta}_p = \frac{V \cos\varphi}{r_p}\dot{\varphi} \tag{H.0.2}$$

where $\dot{\varphi}$ is the target turn rate.

The RMS effect of (H.0.2) on the (angular) position, i.e., the angular displacement over sampling interval T_p (multiplied by 2) is $\ddot{\theta}_p T_p^2$. The (target's true) maneuvering index, with the passive sensor noise SD σ_p (in radians), is the (physically dimensionless) quantity

$$\lambda_p = \frac{\ddot{\theta}_p T_p^2}{\sigma_p} = \frac{\dot{\varphi} T_p^2 V \cos\varphi}{\sigma_p r_p} \tag{H.0.3}$$

For the scenario described in the simulation section, with $V = 250\,\text{m/s}$, $\cos\varphi \approx 0.8$, $r_p \approx 5 \cdot 10^4\,\text{m}$, $T_p = 1\,\text{s}$, $\sigma_p = 1\,\text{mrad}$ and $\dot{\varphi} \approx 30\,\text{mrad/s}$ (which is the maximum target turn rate in our simulation scenario), we have $\lambda_p \approx 0.12$. This small target maneuvering index (less than 0.5) leads to the choice of a KF for the passive sensor, as done in Section 4.5.

Appendix I

Local Estimators' Consistency

The NEES for the active sensor's IMM and for the passive sensor's KF are shown in Figs. I.0.6(a) and I.0.6(b) (the results are obtained by using the estimates and the corresponding covariance matrices for the heterogeneous T2TF, that is, four components' information from the active sensor and two components' information from the passive sensor), respectively. The lack of consistency of the passive sensor KF is due to the maneuvers. The lack of consistency of the active sensor IMM is common and this is due to its (unavoidable delay) in the adaptation. The IMM estimator is "pessimistic" during the no-maneuver intervals and "optimistic" when a maneuver starts or ends until it "catches up". This is the typical behavior of the IMM, which is still superior to any single-model based filter.

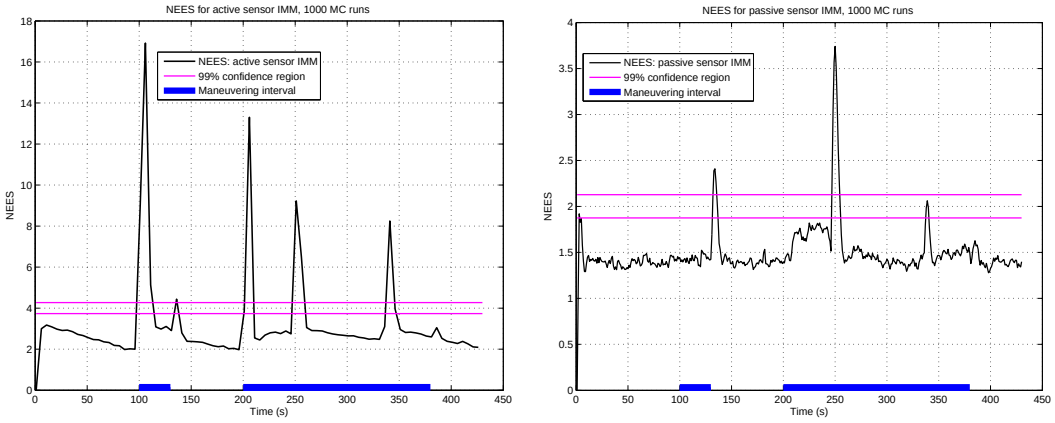


Figure I.0.6: NEES for active sensor IMM and passive sensor KF.

Bibliography

[1] Y. Bar-Shalom and H. Chen, "Covariance Reconstruction for Track Fusion with Legacy Track Sources", *J. of Advances in Information Fusion*, 3(2):107–117, Dec. 2008.

[2] Y. Bar-Shalom, X. R. Li and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction*, Wiley, 2001.

[3] Y. Bar-Shalom, P. K. Willett and X. Tian, *Tracking and Data Fusion*, YBS Publishing, Storrs, CT, Dec. 2011.

[4] D. E. Carlucci and S. S. Jacobson, *Ballistics: Theory and Design of Guns and Ammunition*, CRC Press, 2007.

[5] K.C. Chang, R.K. Saha and Y. Bar-Shalom, "On Optimal Track-to-Track Fusion", *IEEE Trans. Aerosp. Electronic Systems*, 33(4):1271–1276, Oct. 1997.

[6] H. Chen and Y. Bar-Shalom, "Track Association and Fusion with Heterogeneous Local Trackers", *Proc. 46th IEEE Conf. on Decision and Control*, New Orleans, LA, Dec. 2007.

[7] H. Chen, T. Kirubarajan, and Y. Bar-Shalom, "Performance Limits of Track-to-Track Fusion versus Centralized Estimation: Theory and Application", *IEEE Trans. Aerosp. Electronic Systems*, AES-39(2):386–398, Apr. 2003.

[8] A. Farina, L. Timmoneri and D. Vigilante, "Classification and Launch-Impact Point prediction of Ballistic Target via Multiple Model Maximum Likelihood Estimator (MM-MLE)", *Proc. IEEE Radar Conf. '06*, Verona NY, 2006.

[9] D. Goshen-Meskin and I. Y. Bar-Itzhack, "Observability Analysis of Piecewise Constant Systems. I. Theory", *IEEE Trans. Aerosp. Electronic Systems*, 28(4):1056-1067, Oct. 1992.

[10] C. Jauffret, "Observability and Fisher Information Matrix in Nonlinear Regression", *IEEE Trans. Aerosp. Electronic Systems*, 43(2):756-759, April 2007.

[11] T. Kirubarajan and Y. Bar-Shalom, "Kalman filter versus IMM estimator: when do we need the latter?", *IEEE Trans. Aerosp. Electronic Systems*, 39(4):1452–1457, Oct. 2003.

[12] X. R. Li and V. P. Jilkov, "Survey of Maneuvering Target Tracking. Part II: Motion Models of Ballistic and Space Targets", *IEEE Trans. Aerosp. Electronic Systems*, 46(1): 96-119, Feb. 2010.

[13] R. Lopez, P. Danés and F. Royer, "Extending the IMM Filter to Heterogeneous-Order State Space Models", *Proc. 49th IEEE Conf. on Decision and Control*, pp.7369-7374, Atlanta GA, Dec. 2010

[14] R. L. McCoy, *Modern Exterior Ballistics: The Launch and Flight Dynamics of Symmetric Projectiles*, Schiffer Publishing Limited, 1999.

[15] R. K. Mehra, "A Comparison of Several Nonlinear Filters for Reentry Vehicle Tracking", *IEEE Trans. Automatic Control*, AC-16(4): 307-319, Aug. 1971.

[16] M. R. Morelande and N. J. Gordon, "Target tracking through a coordinated turn", *Proc. IEEE International Conf. (ICASSP '05) Acoustics, Speech, and Signal Processing*, Vol(4):iv/21–iv/24, 2005.

[17] T. Nishimura, "On the a Priori Information in Sequential Estimation Problems", *IEEE Trans. Automatic Control*, Vol.11(2): 197-204, Apr. 1966.

[18] R. W. Osborne, III and Y. Bar-Shalom, "Statistical Efficiency of Composite Measurements from LoS Observation", *Proc. SPIE Conf. Signal Processing, Sensor Fusion and Target Recognition*, #8050-07, Orlando FL, April 2011.

[19] R. W. Osborne, III, Y. Bar-Shalom and T. Kirubarajan, "Radar Measurement Noise Variance Estimation with Several Targets of Opportunity", *IEEE Trans. Aerosp. Electronic Systems*, 44(3): 985-995, July 2008.

[20] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes in FORTRAN: The Art of Scientific Computing, 2nd ed*, Cambridge University Press, 1992.

[21] V. Ravindra, Y. Bar-Shalom and P. K. Willett, "Projectile Identification and Impact Point Prediction", *IEEE Trans. Aerosp. Electronic Systems*, 46(4): 2004-2021, Oct. 2010.

[22] F. Reali, G. Palmerini, A. Farina A. Graziano, S. Giompapa and B. Parisi, "Initialization of Ballistic Target Tracking Filters with Detection Probability Lower than Unity", *Proc. 2010 IEEE Aerosp. Conf.*, Big Sky, MT, March 2010.

[23] R. K. Saha "Track-to-Track Fusion With Dissimilar Sensors", *IEEE Trans. Aerosp. Electronic Systems*, 32(3):1021-1029, July 1996.

[24] X. Tian and Y. Bar-Shalom, "Track-to-Track Fusion Configurations and Association in a Sliding Window", *J. of Advances in Information Fusion*, 4(2):146-164, Dec. 2009.

[25] X. Tian and Y. Bar-Shalom, "The Optimal Algorithm for Asynchronous Track-to-Track Fusion", *Proc. SPIE Conf. Signal and Data Processing of Small Targets*, #7698-46, Orlando, FL, April 2010.

[26] Y. C. Xiao, P. Wei and T. Yuan "Observability and Performance Analysis of Bi/Multi-Static Doppler-Only Radar", *IEEE Trans. Aerosp. Electronic Systems*, 46(4):1654-1667, Oct. 2010.

[27] T. Yuan, Y. Bar-Shalom and X. Tian, "Heterogeneous Track-to-Track Fusion", *Proc. 14th International Conf. on Information Fusion*, pp.1-8, Chicago, IL, July 2011.

[28] T. Yuan, Y. Bar-Shalom and X. Tian, "Heterogeneous Track-to-Track Fusion", *J. Advances in Information Fusion*, 6(2): 131-149, 2011.

[29] T. Yuan, Y. Bar-Shalom, P. K. Willett and R. Ben-Dov and S. Pollak, "Estimability of Thrusting Trajectories in 3-D from a Single Passive Sensor", *Proc. SPIE Conf. Signal Processing, Sensor Fusion, and Target Recognition XXII,* #8745-01, Baltimore, MD, April 2013.

[30] T. Yuan, Y. Bar-Shalom, P. K. Willett and R. Ben-Dov and S. Pollak, "Estimability of Thrusting Trajectories in 3-D from a Single Passive Sensor with Unknown Launch Point", *Proc. SPIE Conf. Signal and Data Processing of Small Targets,* #8857-12, San Diego, CA, Aug. 2013.

[31] T. Yuan, Y. Bar-Shalom, P.K. Willett and D. Hardiman, "Impact Point Prediction for Short Range Thrusting Projectiles", *Proc. SPIE Conf. Signal and Data Processing of Small Targets*, #7698-55, Orlando, FL, April 2010.

[32] T. Yuan, Y. Bar-Shalom, P. K. Willett and D. Hardiman, "Impact Point Prediction for Thrusting Projectiles in the Presence of Wind", *Proc. SPIE Conf. Signal Processing, Sensor Fusion, and Target Recognition XXI,* #8392-63, Baltimore, MD, April 2012.

[33] T. Yuan, Y. Bar-Shalom, P. K. Willett and D. Hardiman, "Impact Point Prediction for Thrusting Projectiles in the Presence of Wind", to appear in *IEEE Trans. Aerosp. Electronic Systems*, 2013.

[34] T. Yuan, Y. Bar-Shalom, P. K. Willett, E. Mozeson, S. Pollak and D. Hardiman, "A Multiple IMM Estimation Approach with Unbiased Mixing for

Thrusting Projectiles", *IEEE Trans. Aerosp. Electronic Systems*, 48(4):3250–3267, Oct. 2012.

[35] U.S. Army, "GTRAJ, General Trajectory Program, US Army".

[36] http://www.braeunig.us/space/cd.htm

[37] http://en.wikipedia.org/wiki/External_ballistics